

UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

# **Desarrollo de una red neuronal profunda para la extracción de campos de esfuerzos en imágenes multipolarizadas de fotoelasticidad**

**Diego Eusse Naranjo**

Universidad Nacional de Colombia  
Facultad de Minas, Departamento de Ciencias de la Computación y la Decisión  
Medellín, Colombia  
2023

# **Desarrollo de una red neuronal profunda para la extracción de campos de esfuerzos en imágenes multipolarizadas de fotoelasticidad**

**Diego Eusse Naranjo**

Tesis presentada como requisito parcial para optar al título de:  
**Magíster en Ingeniería - Analítica**

Director: PhD. Alejandro Restrepo Martínez  
Profesor Titular

Línea de Investigación: Ingeniería - Analítica – Redes Neuronales Convolucionales CNNs

Grupos de Investigación:  
Grupo de Promoción e Investigación en Mecánica Aplicada GPIMA

Universidad Nacional de Colombia  
Facultad de Minas, Departamento de Ciencias de la Computación y la Decisión  
Medellín, Colombia  
2023

*En mi pregrado quise aprender a entender y a diseñar máquinas.  
En el posgrado mi deseo es enseñarles a pensar por sí mismas.*

*Diego Eusse Naranjo*

## Declaración de obra original

Yo declaro lo siguiente:

He leído el Acuerdo 035 de 2003 del Consejo Académico de la Universidad Nacional. «Reglamento sobre propiedad intelectual» y la Normatividad Nacional relacionada al respeto de los derechos de autor. Esta disertación representa mi trabajo original, excepto donde he reconocido las ideas, las palabras, o materiales de otros autores.

Cuando se han presentado ideas o palabras de otros autores en esta disertación, he realizado su respectivo reconocimiento aplicando correctamente los esquemas de citas y referencias bibliográficas en el estilo requerido.

He obtenido el permiso del autor o editor para incluir cualquier material con derechos de autor (por ejemplo, tablas, figuras, instrumentos de encuesta o grandes porciones de texto).

Por último, he sometido esta disertación a la herramienta de integridad académica, definida por la universidad.

Diego Eusse Naranjo

Agosto 1 de 2023

## Agradecimientos

A Dios por haberme concedido todos los conocimientos, experiencias y oportunidades para llegar a esta instancia.

A mis padres, Ángela María y Jesús Alberto, por su acompañamiento y su motivación continua para llegar cada vez más lejos.

A mi familia, mi gran soporte, por siempre confiar en mí y formarme en un camino de bien, donde nunca me ha faltado nada.

A Anita, por acompañarme, alentarme y brindarme su amor en este largo camino.

A Jhonatan, Valentina, Rúa y demás amigos y allegados, porque son una gran parte de la motivación y la alegría que se requieren en los momentos más complicados.

A mi director, el profesor Alejandro Restrepo Martínez, por haberme acogido y acompañado todos estos años, sus contribuciones y ejemplo serán siempre parte de mi vida profesional.

A mis compañeros de posgrado Andrés Mauricio Zapata, José Ignacio Amaya, José Miguel Navas, Jaime Andrés Úsuga y Andrés Felipe García, porque siempre es bueno compartir las vivencias y dificultades entre todos.

A los profesores Juan Carlos Briñez de León y John Willian Branch Bedoya, por sus aportes y retroalimentación a este proyecto.

Al profesor Juan David Cano y a todo el equipo de balonmano de la Universidad Nacional Sede Medellín, por enseñarme el significado del compromiso y de un verdadero equipo.

A Pipe, Yuli y Pablo, porque parte de mi día a día y de mi aprendizaje en estos dos años también fue gracias a su apoyo y a sus enseñanzas en el mundo laboral.

A mi querida Alma Máter, la Universidad Nacional de Colombia y en especial a la Facultad de Minas, por todos los conocimientos adquiridos, por formarme como un profesional íntegro, bajo el lema de trabajo y rectitud.

Y a todas las demás personas, quienes de una u otra manera hicieron parte de esta etapa de crecimiento personal, académico y profesional.

¡Muchas gracias!

## Resumen

# Desarrollo de una red neuronal profunda para la extracción de campos de esfuerzos en imágenes multipolarizadas de fotoelasticidad

La fotoelasticidad digital es una técnica experimental utilizada en el análisis y cuantificación de esfuerzos en materiales isótropos y birrefringentes sometidos a una carga, debido a que este tipo de materiales experimentan índices de refracción dobles cuando son cargados, lo que permite estimar la diferencia entre los esfuerzos principales en cada elemento del cuerpo, gracias a un retardo de fase en la luz que lo atraviesa, que puede ser demodulado para extraer un campo de esfuerzos completo.

Tradicionalmente, los procedimientos de extracción de campos de esfuerzos en imágenes de fotoelasticidad eran difíciles de representar con simples regresiones matemáticas, además de que necesitaban montajes costosos, con altos requerimientos de precisión y en ocasiones, algoritmos complicados. En los últimos años, se han desarrollado unas primeras aproximaciones a la automatización de estos procesos por medio del uso de redes neuronales convolucionales para el procesamiento de imágenes isocromáticas en un solo estado de polarización.

En esta tesis, se propone el uso de redes neuronales convolucionales profundas para procesar cuatro imágenes de diferentes estados de polarización. Para ello, se construye una colección de más de setenta mil imágenes analíticas en modelos clásicos de fotoelasticidad, utilizando las propiedades ópticas del material simulado, de la fuente de iluminación y de dos cámaras polarizadas encontradas en la literatura y en cuatro estados de polarización:  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  y  $135^\circ$ .

Estas imágenes son utilizadas para el entrenamiento, validación y testeo de diferentes modelos de red neuronal, basados en arquitecturas clásicas encontradas en la literatura para el desenvolvimiento de fase. Con ello, se realiza un rediseño en las capas, funciones, filtros e hiper-parámetros de las redes estudiadas, en un proceso de optimización de una única arquitectura que alcance las mejores métricas para el problema desarrollado. La evaluación de los modelos se realiza por medio de métricas de calidad de imagen tales como MSE, SSIM, PSNR, entre otras consideradas.

Finalmente, se desarrolla y explica la *MultipolarNet*, una red neuronal especializada en la extracción de campos de esfuerzos en imágenes multipolarizadas de fotoelasticidad. Los resultados abren la posibilidad de procesar imágenes reales generadas por una cámara multipolarizada, lo que representa una gran oportunidad para desarrollar evaluaciones de esfuerzo en tiempo real, predecir problemas de desenvolvimiento de fase en geometrías complejas, cargas dinámicas, o inconsistencias debidas a isoclinicos.

**Palabras Clave:** Fotoelasticidad, Redes Neuronales Convolucionales, Aprendizaje Profundo, Desenvolvimiento de Fase, Análisis de Franjas, Polarización

## Abstract

### Development of a deep neural network for stress field extraction in multipolarized photoelasticity imaging

Digital photoelasticity is an experimental technique used in the analysis and quantification of stresses in isotropic and birefringent materials subjected to a load, due to the fact that this type of materials experience double refractive indices when loaded, which allows estimating the difference between the principal stresses in each element of the body, thanks to a phase delay in the light passing through it, which can be demodulated to extract a complete stress field.

Traditionally, stress field extraction procedures in photoelasticity images were difficult to represent with simple mathematical regressions and required expensive setups with high accuracy requirements and sometimes complicated algorithms. In recent years, first approaches to automate these processes have been developed by using convolutional neural networks for processing isochromatic images in a single polarization state.

In this thesis, we propose the use of deep convolutional neural networks to process four images of different polarization states. For this purpose, a collection of more than seventy thousand analytical images in classical photoelasticity models is built, using the optical properties of the simulated material, the illumination source and two polarized cameras found in the literature and in four polarization states:  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  and  $135^\circ$ .

These images are used for training, validation and testing of different neural network models, based on classical architectures found in the literature for phase unwrapping. With this, a redesign is performed on the layers, functions, filters and general hyper-parameters of the studied networks, in a process of optimization of a single architecture that achieves the best metrics for the developed problem.

The evaluation of the models is performed by means of image quality metrics such as MSE, SSIM, PSNR, among others considered.

Finally, MultipolarNet, a neural network specialized in the extraction of stress fields in multipolarized photoelasticity images, is developed and explained. The results open the possibility of processing real images generated by a multipolarized camera, which represents a great opportunity to develop real-time stress evaluations, predict phase unwrapping problems in complex geometries, dynamic loads, or inconsistencies due to isoclinics.

**Keywords: Photoelasticity, Convolutional Neural Networks, Deep Learning, Phase Unwrapping, Fringe analysis, Polarization**

## Contenido

Declaración de obra original .....	I
Agradecimientos .....	II
Resumen .....	III
Abstract .....	IV
Contenido .....	V
Lista de Imágenes .....	VIII
Lista de Tablas .....	XI
Lista de Ecuaciones.....	XII
Lista de Abreviaturas .....	XIV
1. Introducción .....	1
1.1 Motivación .....	1
1.2 Justificación.....	2
1.3 Preguntas de investigación.....	3
1.4 Hipótesis.....	3
1.5 Objetivos .....	3
1.5.1 Objetivo general .....	3
1.5.2 Objetivos específicos.....	4
1.6 Contribuciones .....	4
1.7 Estructura del documento.....	4
2. Estado del Arte .....	6
2.1 Fotoelasticidad e imágenes multipolarizadas.....	7
2.2 Generación analítica de modelos bajo carga .....	9
2.3 Procesamiento de patrones de franjas mediante DCNNs .....	11
3. Marco Teórico.....	18
3.1 Fotoelasticidad .....	18
3.2 Polariscope.....	19
3.3 Extracción de esfuerzos en imágenes de fotoelasticidad.....	21
3.4 Cámaras e imágenes multipolarizadas .....	23
3.5 Métricas de calidad de imagen .....	23
3.6 Entrenamiento y optimización de redes neuronales.....	25
4. Generación de la colección de imágenes.....	26
4.1 Método .....	26
4.2 Representación de imágenes multipolarizadas.....	35
4.3 Aumento de datos .....	38

4.4	Modelos adicionales para validación .....	41
5.	Desarrollo y Evaluación de Arquitecturas de Redes Neuronales .....	50
5.1	Arquitecturas consideradas .....	50
5.1.1	UNet.....	50
5.1.2	UNet con capas LSTM.....	51
5.1.3	StressNet.....	52
5.1.4	PhotoelastNet.....	53
5.1.5	VGG – Visual Geometry Group .....	53
5.1.6	GAN + Pix2Pix - UNet .....	54
5.2	Implementación de métricas de calidad de imagen .....	55
5.2.1	Tipos de métricas de calidad de imagen.....	56
5.2.2	Métricas tradicionales.....	56
5.2.3	Librería pyiqa.....	56
5.2.4	Métricas adicionales .....	57
5.3	Implementación de arquitecturas.....	57
5.4	Selección de una arquitectura base.....	63
5.4.1	Resultados sobre geometrías conocidas .....	63
5.4.2	Resultados sobre geometrías desconocidas .....	65
5.4.3	Selección.....	68
5.5	Selección de la función de pérdida .....	69
5.5.1	Funciones de pérdida basadas en otros autores .....	69
5.5.2	Funciones de pérdida propias.....	70
5.5.3	Entrenamiento de modelos y resultados.....	71
5.5.4	Selección.....	75
5.6	Optimización de hiper-parámetros.....	75
5.6.1	Keras Tuner.....	76
5.6.2	Experimento para el dropout.....	77
5.6.3	Experimento para el optimizador .....	78
5.6.4	Experimento para las funciones de activación en las capas convolucionales	79
5.6.5	Experimento para el inicializador de kernels.....	81
5.6.6	Optimización conjunta .....	82
5.6.7	Entrenamiento de modelos y resultados.....	83
5.6.8	Selección.....	86
5.7	MultipolarNet .....	86
5.7.1	Características.....	87
5.7.2	Arquitectura .....	87

5.7.3	Predicciones.....	90
5.7.4	Métricas de calidad de imagen .....	96
6.	Conclusiones y trabajo futuro.....	99
7.	Anexos .....	102
7.1	Anexo 1: Interfaz gráfica para el entrenamiento de modelos y evaluación por métricas de calidad de imagen .....	102
7.2	Anexo 2: Código para la generación de la colección de imágenes multipolarizadas.....	103
7.3	Anexo 3: Código base de arquitecturas analizadas .....	103
8.	Referencias .....	104

## Lista de Imágenes

<b>Imagen 1.</b> Línea del tiempo con algunos de los artículos más relevantes del estado del arte. ....	7
<b>Imagen 2.</b> Antecedentes fotoelasticidad e imágenes multipolarizadas. ....	8
<b>Imagen 3.</b> Antecedentes generación analítica de modelos bajo carga. ....	10
<b>Imagen 4.</b> Antecedentes procesamiento de patrones de franjas mediante redes neuronales profundas. ....	12
<b>Imagen 5.</b> Estructura general de una UNet para el desenvolvimiento de fase en un modelo tipo anillo bajo compresión diametral. ....	13
<b>Imagen 6.</b> Representación gráfica de los componentes principales de los esfuerzos. Adaptado de (Briñez et al, 2020) [2]. ....	18
<b>Imagen 7.</b> Esquema de un montaje de polariscopio plano. ....	20
<b>Imagen 8.</b> Esquema de un montaje de polariscopio circular. ....	20
<b>Imagen 9.</b> Esquema de un montaje de polariscopio semicircular. ....	21
<b>Imagen 10.</b> Representación de la polarización lineal de la luz. ....	21
<b>Imagen 11.</b> Representación de la luz a través de un material birrefringente. ....	22
<b>Imagen 12.</b> Apilamiento de cuatro imágenes monocromáticas en diferentes estados de polarización para la generación de una imagen multipolarizada. ....	23
<b>Imagen 13.</b> Respuesta espectral relativa vs Longitud de onda para sensor monocromático IMX250MZR. Adaptado de (IMPERX) [59]. ....	27
<b>Imagen 14.</b> Respuesta espectral relativa vs Longitud de onda para sensor policromático IMX250MZYR. Adaptado de (IMPERX) [59]. ....	28
<b>Imagen 15.</b> Proceso analítico de generación de imágenes en cuatro estados de polarización. Adaptado de Briñez et al., 2020 [12]. ....	32
<b>Imagen 16.</b> Distribución de generación de imágenes en colección inicial. ....	33
<b>Imagen 17.</b> Cuarteto de imágenes por estado de polarización - Geometría de disco a compresión. ....	33
<b>Imagen 18.</b> Cuarteto de imágenes por estado de polarización - Geometría de anillo a compresión. ....	34
<b>Imagen 19.</b> Cuarteto de imágenes por estado de polarización - Geometría de placa infinita. ....	34
<b>Imagen 20.</b> Cuarteto de imágenes por estado de polarización - Geometría de viga carga distribuida. ....	34
<b>Imagen 21.</b> Cuarteto de imágenes por estado de polarización - Geometría de viga carga puntual. ....	35
<b>Imagen 22.</b> Generación de imágenes multipolarizadas por apilamiento de imágenes ( <i>Stack</i> ). ....	36
<b>Imagen 23.</b> Representación por apilamiento de imágenes ( <i>Stack</i> ) en diferentes geometrías. ....	36
<b>Imagen 24.</b> Generación de imágenes multipolarizadas por Vector Stokes. ....	37
<b>Imagen 25.</b> Representación por vector Stokes en diferentes geometrías. ....	38
<b>Imagen 26.</b> Distribución total de imágenes por colección, con proceso de aumento de datos. ....	39
<b>Imagen 27.</b> Alteración por ruido gaussiano en la colección de imágenes apiladas ( <i>Stack</i> ). ....	39
<b>Imagen 28.</b> Alteración por ruido gaussiano en la colección de imágenes Stokes. ....	40
<b>Imagen 29.</b> Alteración por brillo y contraste en la colección de imágenes apiladas ( <i>Stack</i> ). ....	40
<b>Imagen 30.</b> Alteración por brillo y contraste en la colección de imágenes Stokes. ....	40

<b>Imagen 31.</b> Alteración por rotaciones en la colección de imágenes apiladas ( <i>Stack</i> ). ....	41
<b>Imagen 32.</b> Alteración por rotaciones en la colección de imágenes Stokes. ....	41
<b>Imagen 33.</b> Alteración por recortes y canales en la colección de imágenes apiladas ( <i>Stack</i> ). .....	41
<b>Imagen 34.</b> Alteración por recortes y canales en la colección de imágenes Stokes. ....	41
<b>Imagen 35.</b> Diagrama de modelo de flexión de viga sometida a carga variable continua. Adaptado de Timoshenko et al., 1951 [37]. ....	42
<b>Imagen 36.</b> Cuarteto de imágenes por estado de polarización - Geometría de estructura vertical bajo presión continua. ....	43
<b>Imagen 37.</b> Simulación de estructura vertical bajo presión lateral en Ansys Workbench.	43
<b>Imagen 38.</b> Diagrama de modelo de ménsula cargada en extremo libre. Adaptado de Timoshenko et al., 1951 [37]. ....	44
<b>Imagen 39.</b> Cuarteto de imágenes por estado de polarización - Geometría de ménsula curva cargada en su extremo libre. ....	45
<b>Imagen 40.</b> Simulación de ménsula curva cargada en su extremo libre en Ansys Workbench. ....	45
<b>Imagen 41.</b> Diagrama de modelo de placa rectangular finita con muesca tipo 'V' bajo compresión uniaxial. Adaptado de Kramer et al., 2009 [38]. ....	46
<b>Imagen 42.</b> Cuarteto de imágenes por estado de polarización - Geometría de placa rectangular finita con muesca tipo 'V' bajo compresión uniaxial. ....	47
<b>Imagen 43.</b> Simulación de placa rectangular finita con muesca tipo 'V' bajo compresión uniaxial en Ansys Workbench. ....	47
<b>Imagen 44.</b> Representación por apilamiento de imágenes ( <i>Stack</i> ) en geometrías de validación. ....	48
<b>Imagen 45.</b> Alteraciones para aumento de datos en la colección de imágenes de validación por representación apilada ( <i>Stack</i> ). ....	49
<b>Imagen 46.</b> Esquema UNet clásica adaptada al problema de extracción de esfuerzos en imágenes multipolarizadas. Adaptado de Perera et al., 2020 [14]. ....	51
<b>Imagen 47.</b> Esquema UNet con capas LSTM adaptada al problema de extracción de esfuerzos en imágenes multipolarizadas. Adaptado de Perera et al., 2020 [14]. ....	52
<b>Imagen 48.</b> Esquema StressNet y componentes. Adaptado de Briñez et al., 2020 [3]. ...	53
<b>Imagen 49.</b> Esquema PhotoelastNet y componentes. Adaptado de Briñez et al., 2022 [2]. .....	53
<b>Imagen 50.</b> Bloques que conforman la arquitectura VGG. Adaptado de Tao et al., 2022 [4]. .....	54
<b>Imagen 51.</b> Esquema GAN + Pix2Pix y capas componentes. Adaptado de Park et al., 2022 [46]. ....	55
<b>Imagen 52.</b> Resultados sobre geometría de disco a compresión para las arquitecturas estudiadas. ....	59
<b>Imagen 53.</b> Resultados sobre geometría de anillo a compresión para las arquitecturas estudiadas. ....	59
<b>Imagen 54.</b> Resultados sobre geometría de placa infinita para las arquitecturas estudiadas. .....	60
<b>Imagen 55.</b> Resultados sobre geometría de viga bajo carga puntual para las arquitecturas estudiadas. ....	60
<b>Imagen 56.</b> Resultados sobre geometría de estructura vertical bajo presión lateral para las arquitecturas estudiadas. ....	61
<b>Imagen 57.</b> Resultados sobre geometría de ménsula curva cargada en su extremo libre para las arquitecturas estudiadas. ....	61

<b>Imagen 58.</b> Resultados sobre geometría de placa rectangular finita con muesca tipo 'V' bajo compresión uniaxial para las arquitecturas estudiadas. ....	62
<b>Imagen 59.</b> Métrica del error cuadrático medio (MSE) para la colección de geometrías conocidas, por arquitectura.....	63
<b>Imagen 60.</b> Métrica del Índice de similitud estructural (SSIM) para la colección de geometrías conocidas, por arquitectura. ....	64
<b>Imagen 61.</b> Métrica de la Proporción máxima de señal a ruido (PSNR) para la colección de geometrías conocidas, por arquitectura. ....	65
<b>Imagen 62.</b> Métrica del error cuadrático medio (MSE) para la colección de geometrías desconocidas, por arquitectura. ....	66
<b>Imagen 63.</b> Métrica del Índice de similitud estructural (SSIM) para la colección de geometrías desconocidas, por arquitectura. ....	67
<b>Imagen 64.</b> Métrica de la Proporción máxima de señal a ruido (PSNR) para la colección de geometrías desconocidas, por arquitectura. ....	68
<b>Imagen 65.</b> Resultados sobre geometría de estructura vertical bajo presión lateral para las funciones de pérdida estudiadas. ....	72
<b>Imagen 66.</b> Resultados sobre geometría de ménsula curva cargada en su extremo libre para las funciones de pérdida estudiadas. ....	72
<b>Imagen 67.</b> Resultados sobre geometría de placa rectangular finita con muesca tipo 'V' bajo compresión uniaxial para las funciones de pérdida estudiadas.....	73
<b>Imagen 68.</b> Métrica del error cuadrático medio (MSE) para la colección de geometrías desconocidas, por función de pérdida.....	74
<b>Imagen 69.</b> Métrica del Índice de similitud estructural (SSIM) para la colección de geometrías desconocidas, por función de pérdida. ....	74
<b>Imagen 70.</b> Métrica de la Proporción máxima de señal a ruido (PSNR) para la colección de geometrías desconocidas, por función de pérdida. ....	75
<b>Imagen 71.</b> Variables de optimización del modelo de red neuronal. ....	76
<b>Imagen 72.</b> Predicciones sobre geometría de estructura vertical en modelos optimizados con Keras. ....	83
<b>Imagen 73.</b> Predicciones sobre geometría de ménsula en modelos optimizados con Keras. ....	83
<b>Imagen 74.</b> Predicciones sobre geometría de placa con muesca en V en modelos optimizados con Keras.....	84
<b>Imagen 75.</b> Métrica del error cuadrático medio (MSE) para la colección de geometrías desconocidas, por modelo optimizado. ....	84
<b>Imagen 76.</b> Métrica del Índice de similitud estructural (SSIM) para la colección de geometrías desconocidas, por modelo optimizado. ....	85
<b>Imagen 77.</b> Métrica de la Proporción máxima de señal a ruido (PSNR) para la colección de geometrías desconocidas, por modelo optimizado. ....	86
<b>Imagen 78.</b> Representación bloques encoder y decoder de la MultipolarNet. ....	89
<b>Imagen 79.</b> Representación de la arquitectura de la MultipolarNet a partir de los bloques encoder y decoder definidos en la imagen 78. ....	90
<b>Imagen 80.</b> Interfaz gráfica para entrenamiento de redes y demodulación de imágenes multipolarizadas.....	102

## Lista de Tablas

<b>Tabla 1.</b> Características generales de arquitecturas consideradas.....	58
<b>Tabla 2.</b> Valores del experimento para la optimización del dropout.....	78
<b>Tabla 3.</b> Valores del experimento para la optimización del optimizador.....	79
<b>Tabla 4.</b> Valores del experimento para la optimización de las funciones de activación....	80
<b>Tabla 5.</b> Valores del experimento para la optimización de los inicializadores de kernels.....	82
<b>Tabla 6.</b> Valores del experimento para la optimización conjunta.....	82
<b>Tabla 7.</b> Resultados de Keras Tuner para la optimización conjunta.....	83
<b>Tabla 8.</b> Características de la MultipolarNet.....	87
<b>Tabla 9.</b> Arquitectura MultipolarNet.....	87
<b>Tabla 10.</b> Predicciones de la MultipolarNet ante cargas altas.....	90
<b>Tabla 11.</b> Predicciones de la MultipolarNet ante cargas medias.....	91
<b>Tabla 12.</b> Predicciones de la MultipolarNet ante cargas bajas.....	91
<b>Tabla 13.</b> Predicciones de la MultipolarNet ante alteraciones de ruido.....	92
<b>Tabla 14.</b> Predicciones de la MultipolarNet ante alteraciones de brillo y contraste.....	93
<b>Tabla 15.</b> Predicciones de la MultipolarNet ante alteraciones de rotaciones.....	93
<b>Tabla 16.</b> Predicciones de la MultipolarNet ante alteraciones de recortes.....	94
<b>Tabla 17.</b> Predicciones de la MultipolarNet con diferentes fuentes de iluminación.....	95
<b>Tabla 18.</b> Predicciones de la MultipolarNet con diferentes sensores de cámara.....	95
<b>Tabla 19.</b> Promedio y desviación estándar por iteración de la MultipolarNet en métricas de calidad de imagen tradicionales sobre geometrías desconocidas.....	96
<b>Tabla 20.</b> Promedio y desviación estándar por iteración de la MultipolarNet en métricas de calidad de imagen “Full Reference” (FR) sobre geometrías desconocidas.....	97
<b>Tabla 21.</b> Promedio y desviación estándar por iteración de la MultipolarNet en métricas de calidad de imagen “No Reference” (FR) sobre imágenes desconocidas. (Parte I).....	97
<b>Tabla 22.</b> Promedio y desviación estándar por iteración de la MultipolarNet en métricas de calidad de imagen “No Reference” (FR) sobre imágenes desconocidas. (Parte II).....	98

## Lista de Ecuaciones

Ecuación	Nombre
(1)	Campo eléctrico de la luz en función de amplitud, tiempo y frecuencia angular de la onda.
(2)	Retardo relativo en la componente de campo eléctrico en función del retardo de la onda y la longitud de onda
(3)	Retardo de fase en función del espesor de un material e índices de refracción
(4)	Retardo de fase en función del espesor de un material, coeficiente óptico y diferencia de esfuerzos principales
(5) y (6)	Esfuerzos principales a partir del retardo de fase y del conteo de franjas
(7)	Fórmula del Error Cuadrático Medio, MSE
(8)	Fórmula de la Proporción Máxima de Señal a Ruido, PSNR
(9)	Fórmula de la Medida del Índice de Similitud Estructural, SSIM
(10) a (12)	Ecuaciones de la distribución de esfuerzos en un disco a compresión
(13) a (22)	Ecuaciones de la distribución de esfuerzos en un anillo a compresión
(23) a (26)	Ecuaciones de la distribución de esfuerzos en una placa infinita con perforación central
(27) a (30)	Ecuaciones de la distribución de esfuerzos en una viga rectangular de espesor unitario en voladizo y carga distribuida transversal
(31) a (33)	Ecuaciones de la distribución de esfuerzos en una viga rectangular de espesor unitario en voladizo y carga puntual transversal
(34)	Ecuaciones de Stokes para formar $S_0$ , $S_1$ y $S_2$ en imágenes polarizadas
(35) a (37)	Ecuaciones de la distribución de esfuerzos en una estructura vertical a flexión por presión continua
(38) a (44)	Ecuaciones de la distribución de esfuerzos en una ménsula curva cargada en su extremo libre
(39) a (47)	Ecuaciones de la distribución de esfuerzos en una placa rectangular finita con muesca tipo "V" bajo compresión uniaxial
(48)	Función de pérdida de Perera et al., 2020 [14]
(51) a (53)	Función de pérdida de Tao et al., 2022 [4]
(54) y (55)	Función de pérdida a partir de MSE, MAE y SSIM ponderados
(56)	Función de pérdida a partir de las métricas MAD, BPE y SSIM

- (57) | Función de pérdida a partir de los errores MSE, MAE y las métricas MAD, BPE y SSIM

## Lista de Abreviaturas

Abreviatura	Significado
DCNNs	Deep Convolutional Neural Networks
CNNs	Convolutional Neural Networks
LUT	Look Up Table
PST	Phase Shifting Technique
PSN	Phase Shifting Network
ML	Machine Learning
DL	Deep Learning
GAN	Generative Adversarial Network
MSE	Mean Squared Error
MAE	Mean Absolute Error
SSIM	Structural Similarity Index Measure
PSNR	Peak Signal-to-Noise Ratio
NR	No-Reference
FR	Full-Reference
BRISQUE	Blind Referenceless Image Spatial Quality Evaluator
DBCNN	Deep Bilinear Convolutional Neural Network
NIMA	Neural Image Assessment
NRQM	No-Reference Quality Metric
CKDN	Conditional Knowledge Distillation Network
DISTS	Deep Image Structure and Texture Similarity
LPIPS	Learned Perceptual Image Patch Similarity
SGD	Stochastic Gradient Descent
RGB	Red, Green and Blue Image Channels
VGG	Visual Geometry Group Neural Network
ReLU	Rectified Linear Unit
ELU	Exponential Linear Unit
SELU	Scaled Exponential Linear Unit

# 1. Introducción

En este capítulo se realiza una descripción del fenómeno de fotoelasticidad como técnica óptica y su aplicación en el análisis de esfuerzos. Se realiza una descripción sobre su relación con el aprendizaje profundo, las redes neuronales convolucionales y su utilidad como procedimiento alternativo, preciso y eficiente para la demodulación de las imágenes de fotoelasticidad digital en campos de esfuerzos. Posteriormente se enuncian las preguntas de investigación, hipótesis, objetivos, contribuciones y estructura de esta tesis.

## 1.1 Motivación

La fotoelasticidad digital es una técnica óptica experimental utilizada en el análisis y la cuantificación de esfuerzos en materiales isótropos y birrefringentes sometidos a una carga.

El fenómeno de fotoelasticidad podría resumirse de la siguiente manera: al aplicar una carga sobre un material birrefringente, la fuerza aplicada en el material produce una distribución de esfuerzos en la geometría, lo que genera una variación en el índice de refracción del material, que al ubicarse debidamente dentro de un montaje óptico tipo polariscopio (involucrando fuentes de iluminación, retardadores de cuarto de onda y polarizadores) permite la visualización de patrones de franjas isocromáticas e isoclínicas, las cuales describen la diferencia de esfuerzos principales y su dirección, respectivamente.

La demodulación de estos patrones de franjas se ha realizado históricamente por medio de diferentes métodos matemáticos, que implicaban procedimientos de conteo manual de franjas y montajes ópticos costosos que requieren de una alta precisión, al posicionar y calibrar los diferentes elementos ópticos dentro del polariscopio. (Briñez et al., 2013) [1]

Con la finalidad de buscar otras técnicas alternativas que evitaran estas problemáticas, algunos procesos de demodulación de patrones de franjas isocromáticas se han implementado a través de los años. Dentro de estas alternativas vale la pena mencionar a los procedimientos de métodos de cambio de fase (Phase Shifting), técnicas de paso de carga (Load Stepping), comparaciones de color por fotoelasticidad de 12 franjas, seguimiento de fase regularizado (Phase Tracking), estrategias híbridas computacionales, algoritmos basados en evolución del color, y otros métodos. (Briñez et al., 2022) [2]

Adicional a esto y debido a la gran variedad de posibilidades involucradas en la generación de imágenes de un fenómeno de fotoelasticidad (condiciones e iluminación, ruido, posicionamiento de elementos ópticos, sensores en las cámaras, polarización, entre otros) y a los tamaños de las colecciones de imágenes que podrían construirse con dichas variedades, se ha optado en los últimos años por entrenar algoritmos de aprendizaje automático y aprendizaje profundo para realizar la extracción de campos de esfuerzos.

Los métodos de aprendizaje profundo son reconocidos por sus importantes contribuciones en áreas que a menudo eran difíciles de representar con simples regresiones matemáticas (Briñez et al., 2020) [3] o que implicaban costosos experimentos y complicados algoritmos. Dentro de este tipo de tareas mejoradas, es posible encontrar la eliminación de ruido de fase, el desenvolvimiento de fase, la reconstrucción 3D, la segmentación de objetos, entre otras tareas (Wang et al., 2019) [8].

En este contexto, la presente tesis genera un modelo de regresión de imágenes de fotoelasticidad, el cual toma patrones de franjas en imágenes multipolarizadas y retorna una única imagen que representa el campo de esfuerzos presente en la geometría cargada.

Debido a la gran variedad de condiciones que pueden alterar a cada una de estas imágenes y a la cantidad de imágenes que ello conlleva, se decide enfocar al problema en la construcción de un algoritmo tipo red neuronal, que se entrena por medio de una colección inicial de imágenes analíticas, con el objetivo de crear un modelo generalizable, capaz de predecir en geometrías más complejas o bien, en imágenes reales.

## 1.2 Justificación

La generación correcta de imágenes de fotoelasticidad depende en gran medida de la precisión y de las condiciones con las que se realice el montaje del polariscopio y puede generar un costo elevado debido a los elementos ópticos involucrados. Específicamente, debe garantizarse una coordinación en el posicionamiento de los elementos ópticos, tales como polarizadores y retardadores de cuarto de onda, para lograr una visualización adecuada de los patrones de franjas generados en la geometría cargada y evitar con ello la aparición de posibles defectos generados por isoclínicos, que se representan como discontinuidades en las imágenes con patrones de franjas.

Adicional a esto, es importante mencionar que incluso si se contara con un montaje de polariscopio preciso, en el que los patrones isoclínicos pueden ser atenuados al garantizar la correcta configuración del ángulo entre los ejes de los polarizadores, la extracción de los esfuerzos aún requeriría del cálculo del retardo de fase y del conteo manual de las franjas. (Briñez et al., 2013) [1].

Por lo cual, es posible afirmar que la recuperación de la información sobre el esfuerzo de estos patrones sigue siendo un reto, debido a la necesidad de algoritmos de desenvolvimiento capaces de extraer la información envuelta en estos patrones de franjas, de una manera más simple y rápida.

Además es de gran importancia mencionar que, si bien es posible encontrar algunas implementaciones recientes de aprendizaje profundo en las que se extraen campos de esfuerzos a partir de imágenes de fotoelasticidad, como las realizadas por (Briñez et al., 2022) [2] y otros autores más, estas implementaciones se limitan únicamente al uso de imágenes isocromáticas y solo se enfocan en imágenes de un solo estado de polarización, omitiendo algunos procedimientos de fotoelasticidad por medio de cámaras multipolarizadas, como el desarrollado por (Ren et al., 2022) [5].

Este tipo de cámaras permiten la obtención de hasta 4 imágenes dentro de una misma captura, cada una con una orientación diferente del polarizador que hace parte de ellas (0°, 45°, 90° y 135°). Esto permite que los procesos sean más precisos al tener más información del objeto de estudio.

La tesis presentada en este documento plantea el desarrollo de un algoritmo tipo red neuronal para la extracción de campos de esfuerzos en imágenes multipolarizadas de fotoelasticidad.

Para ello, se generará inicialmente una colección amplia y refinada de imágenes analíticas y multipolarizadas, con las que se entrenarán diferentes modelos teniendo en cuenta las diferentes variaciones y alteraciones posibles en la generación de imágenes de fotoelasticidad. Con ello, se evaluarán posteriormente cada uno de los modelos de aprendizaje profundo planteados y se escogerá el modelo que presente mejores métricas de calidad de imagen en el proceso de extracción de esfuerzos a través de la demodulación.

### **1.3 Preguntas de investigación**

Con base en la información descrita anteriormente y en los antecedentes encontrados en la literatura, se plantearon las siguientes preguntas de investigación para esta tesis:

- ¿Es posible extraer campos de esfuerzos, utilizando redes neuronales profundas y partiendo de imágenes multipolarizadas de fotoelasticidad digital?
- Si se utilizan diferentes funciones de pérdida y métricas de calidad de imagen, ¿cuál de las arquitecturas de red neuronal desarrolladas, alcanza un desempeño más adecuado para la problemática estudiada?
- ¿El modelo de red seleccionado es capaz de extraer campos de esfuerzos en imágenes multipolarizadas que presenten isoclínicos y es capaz de generalizar ante imágenes de fotoelasticidad bajo diferentes condiciones?

### **1.4 Hipótesis**

Continuando con las preguntas de investigación planteadas previamente, se propone la siguiente hipótesis para este proyecto de investigación:

“Las redes neuronales profundas pueden extraer campos de esfuerzos en materiales birrefringentes sometidos a carga, partiendo de imágenes multipolarizadas de fotoelasticidad digital, que incluyen (o no) la presencia de defectos debido a isoclínicos, las diferentes condiciones de las imágenes y evaluando los resultados mediante métricas de calidad de imagen.”

### **1.5 Objetivos**

Como parte del planteamiento y desarrollo de este proyecto de investigación se plantean los siguientes objetivos:

#### **1.5.1 Objetivo general**

Desarrollar un modelo de red neuronal profunda para la extracción de campos de esfuerzos partiendo de imágenes multipolarizadas de fotoelasticidad.

## 1.5.2 Objetivos específicos

1. Implementar una estrategia de generación de imágenes digitales que represente las diferentes condiciones reales de los modelos de fotoelasticidad.
2. Establecer un método para determinar los hiper-parámetros de las arquitecturas de red neuronal.
3. Evaluar el desempeño de las diferentes arquitecturas de red neuronal utilizando métricas de calidad de imagen.

## 1.6 Contribuciones

Se adaptaron y desarrollaron varias arquitecturas de red neuronal para la extracción de campos de esfuerzos en imágenes multipolarizadas de fotoelasticidad. Entre ellas, se destaca el desarrollo y evaluación de la *MultipolarNet*, la red neuronal que reportó las mejores métricas de calidad de imagen dentro de lo experimentado por esta tesis.

Esta tesis también permitió la participación en dos congresos de carácter internacional por medio de ponencias tipo póster y artículos científicos:

1. El artículo *Stress Fields Extraction in Multi-Polarized Photoelasticity Images Using Deep Convolutional Neural Networks*, publicado en el evento *Imaging and Applied Optics Congress 2022 (COSI)* de la Sociedad Óptica Estadounidense – OSA, puede ser encontrado [aquí](#).
2. El artículo *Photoelastic and Stokes Images through Deep Convolutional Neural Networks: A Comparison of Stress Fields*, publicado en el evento *Optical Metrology Symposium*, de la Sociedad de Ingenieros de Instrumentación Foto-Óptica – SPIE, puede ser encontrado [aquí](#).

Adicional a esto, se implementó una interfaz gráfica capaz de entrenar y ejecutar predicciones sobre imágenes multipolarizadas de fotoelasticidad, utilizando varias de las arquitecturas experimentadas en esta tesis. El código fuente de este software puede ser encontrado en el Anexo 1.

## 1.7 Estructura del documento

Este documento está organizado de la siguiente manera: Se presenta la introducción, justificación y estructura de la tesis en el capítulo 1. El capítulo 2 muestra los antecedentes y el estado del arte consultado para el desarrollo de la tesis. En el capítulo 3 se proporciona un marco teórico con las temáticas necesarias para el entendimiento del problema. A partir de allí, se desarrollan los objetivos específicos mencionados previamente, en los capítulos 4 y 5: Capítulo 4 toma la generación de una colección de imágenes multipolarizadas y capítulo 5 explica todo el proceso de entrenamiento, validación y optimización de la red neuronal. Por su parte, el capítulo 6 enuncia las conclusiones y trabajo futuro propuesto.

Adicionalmente en el capítulo 7, se agregan algunos anexos elaborados en el desarrollo de la tesis y que podrían servir de soporte para sustentar algunas de las conclusiones halladas; o bien, para fundamentar algunas investigaciones futuras.

Finalmente, en el capítulo 8 se listan las referencias y la bibliografía utilizada en la construcción de esta tesis.

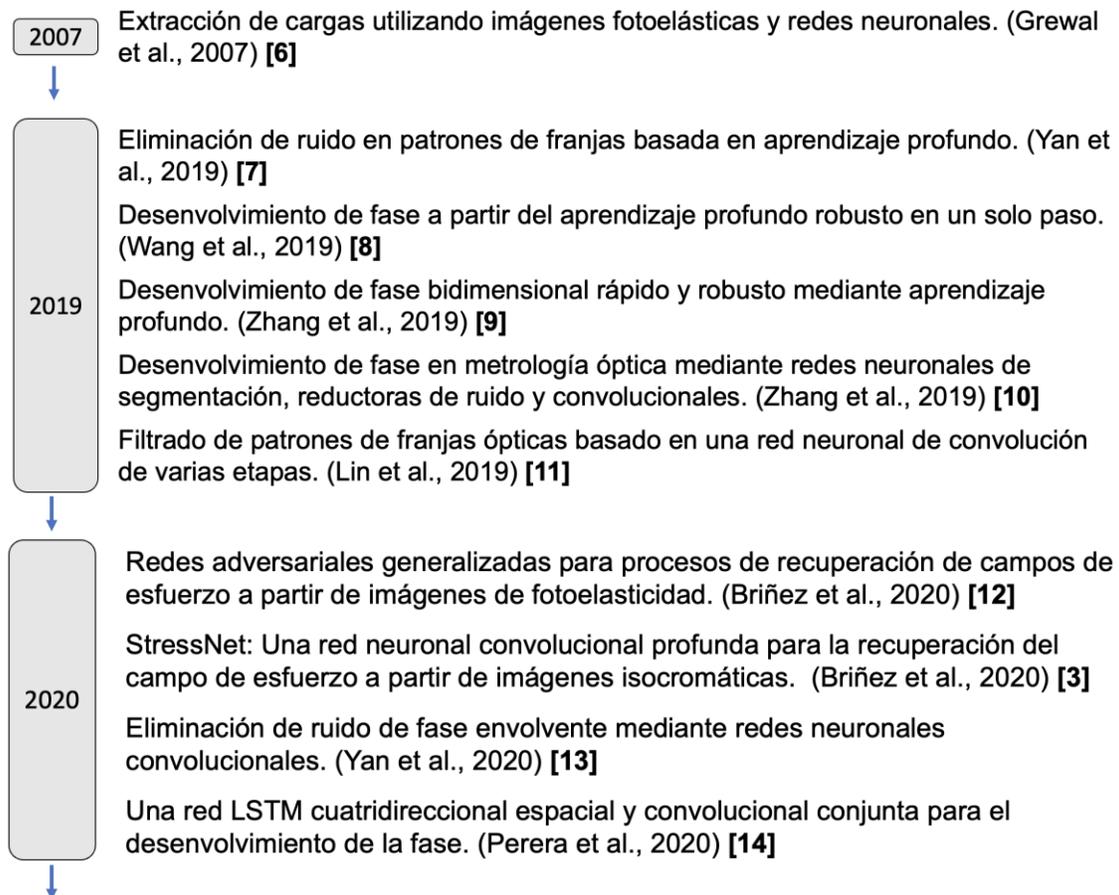
## 2. Estado del Arte

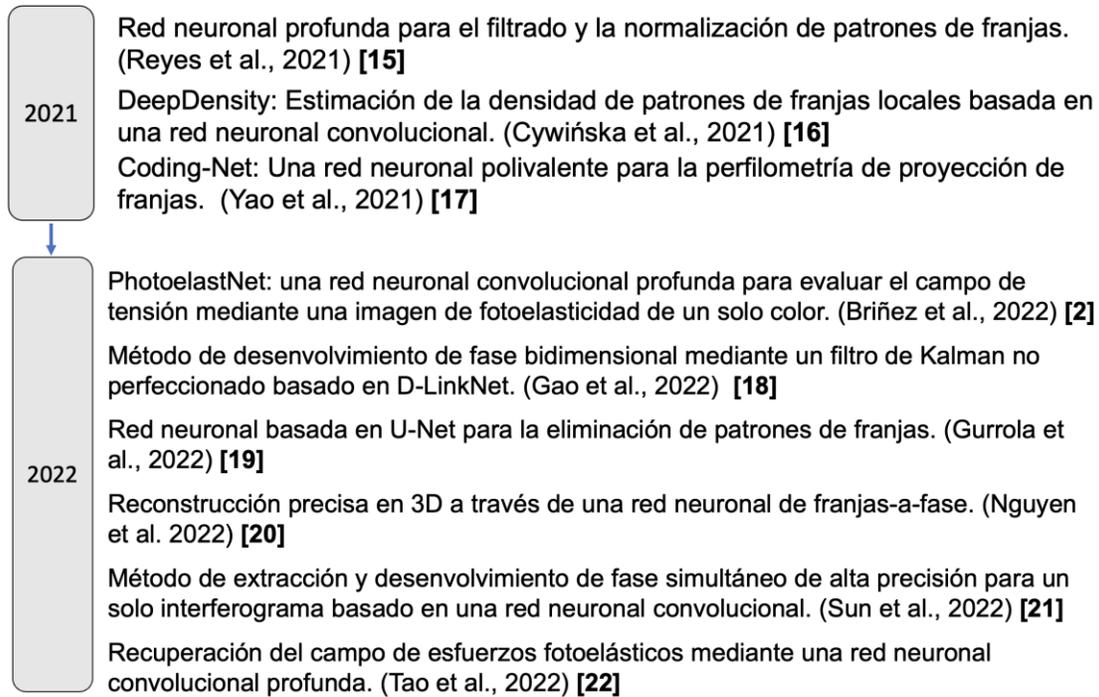
En este capítulo se presentan los antecedentes y la revisión del estado del arte para el problema desarrollado.

El estado del arte realizado se compone de aproximadamente 50 artículos, con un enfoque investigativo hacia los datos y su adquisición, las metodologías usadas, los modelos predictivos y arquitecturas de redes, las oportunidades que quedan latentes y las conclusiones a las que se llega en cada estudio.

Se decide organizar los resultados encontrados mediante una línea de tiempo que permite observar el avance de estas temáticas desde el resultado más antiguo (2007), hasta los más modernos al momento de la revisión del estado del arte (2023).

Por temas prácticos se decide enunciar aquellos artículos con un aporte significativo a la tesis desarrollada. La imagen 1 ilustra la línea del tiempo construida con los artículos más relevantes:





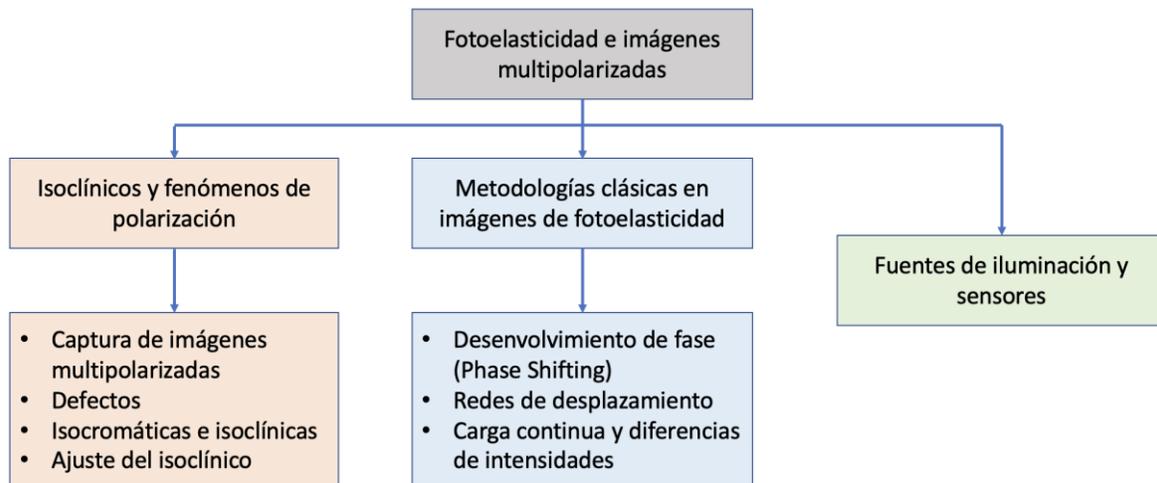
**Imagen 1.** Línea del tiempo con algunos de los artículos más relevantes del estado del arte.

A continuación, se profundizan los resultados, conclusiones, recomendaciones y problemáticas encontradas en estos artículos, dividiéndolos en tres categorías, de acuerdo con la temática analizada, de la siguiente manera: Sección 2.1: Fotoelasticidad e imágenes multipolarizadas. Sección 2.2: Generación analítica de imágenes de fotoelasticidad en modelos bajo carga. Sección 2.3: Procesamiento de patrones de franjas mediante redes neuronales profundas y métricas de calidad de imagen.

## 2.1 Fotoelasticidad e imágenes multipolarizadas

Dentro de esta primera temática en la que se enfocó la revisión de antecedentes, se encontraron principalmente resultados sobre defectos ocasionados por isoclínicos, estudios que involucran el uso de cámaras multipolarizadas, metodologías clásicas de procesamiento de imágenes de fotoelasticidad, diferentes a los procesamientos mediante redes neuronales y algunos estudios enfocados en las fuentes de iluminación y sensores.

La imagen 2 ilustra los hallazgos dentro de esta temática.



**Imagen 2.** Antecedentes fotoelasticidad e imágenes multipolarizadas.

Son varios los autores que involucran el estudio de patrones de franjas con isoclínicos dentro de sus temáticas. En su artículo, Ren et al., 2020 [5] utilizan una cámara polarizada que captura simultáneamente los patrones foto-elásticos de campos oscuros y brillantes bajo un polariscopio circular, propone un método para cuantificar los órdenes de franjas, en un disco bajo carga a compresión y compara los órdenes de franja calculados por los métodos tradicional y un nuevo método propuesto, en estructuras complejas. Sin embargo, existen aún desafíos y oportunidades en cuanto a la eliminación de la influencia de la luz de fondo y a la determinación de la relación existente entre la precisión de desenvolvimiento y la tasa de captura de imágenes.

También es valioso destacar el artículo desarrollado por Ramesh et al., 2020 [23], en el cual se ilustran los posibles defectos que pueden presentarse en una imagen con patrones de franjas debido a la presencia de isoclínicos, esto es, la presencia de un ángulo entre los polarizadores del polariscopio, que impide que el campo de esfuerzos se muestre completamente y genera defectos tanto en las imágenes isocromáticas (que representan la diferencia de esfuerzos principales) y en las isoclínicas (que representan la dirección en que se generan dichos esfuerzos). Los defectos en el campo isoclínico son denominados *inconsistencias*, mientras que los del campo isocromático son denominados *ambigüedades*.

Adicional a esto, Shang et al., 2022 [24] demostraron que los parámetros isoclínicos e isocromáticos se afectan mutuamente y basándose en el análisis de la interacción, propusieron un algoritmo para ajustar la mutación del parámetro isoclínico. Este método demostró ser capaz de eliminar eficazmente la influencia del parámetro isocromático para obtener el parámetro isoclínico correcto afectado por el parámetro isocromático.

En cuanto a las metodologías clásicas de procesamiento de imágenes de fotoelasticidad, Briñez et al., 2018 [25] desarrollaron un método híbrido computacional para reducir la etapa de adquisición en una técnica de desenvolvimiento de fase (PST) convencional, demostraron que los mapas isoclínicos e isocromáticos pueden ser evaluados mediante PST convencional realizando menos adquisiciones, e introduciendo operaciones computacionales dentro del proceso. Sin embargo, los resultados indicaron que los usuarios deben tener en cuenta la distribución espacial en la fuente de luz y que existen limitaciones

para recuperar la información de esfuerzos en zonas en las que se aplica la carga. Este efecto se debe a la alta densidad de franjas generada por las altas concentraciones de esfuerzo.

También, resulta valioso enunciar los resultados encontrados por Rodríguez et al., 2017 [26] y por Zhang et al., 2021 [27], en los cuales se implementan métodos alternativos para la demodulación de patrones de franjas cerradas mediante corrimiento de fase y mediante redes de desplazamiento de fase (PSN) para mapear la relación entre interferogramas, respectivamente.

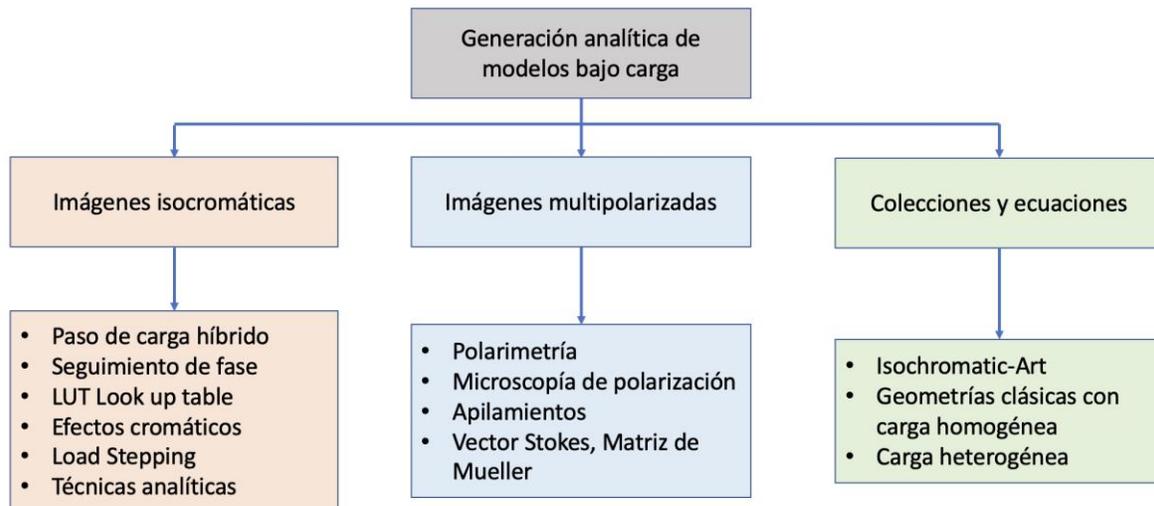
Además de los métodos tradicionales de corrimiento y desenvolvimiento de fase, Ren et al., 2020 [5] implementó también un método de cuantificación de los órdenes de franja de campo completo basado en la diferencia de intensidad luminosa entre estos patrones y utilizando carga continua. Este método eliminaba la influencia de la luz de fondo en la cuantificación de los órdenes de las franjas. El análisis de precisión indicó que este método era aplicable para determinar los órdenes de franja de campo completo.

Finalmente, en el análisis de fuentes de iluminación y sensores de cámaras, resulta bastante importante el trabajo realizado por Fandiño et al., 2023 [28], en el cual se desarrolló un análisis computacional del efecto causado por la selección de la fuente de luz y el sensor de la cámara para experimentos de fotoelasticidad. La estrategia mostró que la calidad de los patrones de franjas, que representan la información del esfuerzo, se ve afectada por el tipo de fuente de luz y sensor de la cámara, y que este problema se traslada a la calidad con la que se recupera el mapa de esfuerzos. Lo cual podría indicar que para los experimentos de fotoelasticidad debería llevarse a cabo una cuidadosa selección de la fuente de luz y del sensor de la cámara.

## 2.2 Generación analítica de modelos bajo carga

La segunda temática de búsqueda de información y antecedentes responde especialmente al objetivo específico 1, pues se desea generar una colección de imágenes digitales que represente las diferentes condiciones reales de los modelos de fotoelasticidad, con la cual sea posible llevar a cabo el entrenamiento de las diferentes arquitecturas de red neuronal y su posterior evaluación. Por ello, resulta fundamental realizar una búsqueda de colecciones de imágenes públicas, multipolarizadas y de fotoelasticidad digital; o bien, de estudios que involucren ecuaciones o simulaciones que puedan guiar la generación de una colección propia.

La imagen 3 resume los hallazgos dentro de esta temática.



**Imagen 3.** Antecedentes generación analítica de modelos bajo carga.

En los últimos años, algunos estudios se han enfocado en imágenes isocromáticas en un único estado de polarización y otros más se han centrado en el estudio de imágenes multipolarizadas.

En el caso de las imágenes isocromáticas, se han utilizado estrategias de una sola captura basadas en el paso de carga híbrido (Briñez et al., 2018) [29], el seguimiento de fase regularizado (Quiroga et al., 2020) [30] y la comparación de Look Up Table (LUT) (Pandey et al., 2019) [31]. Sin embargo, sigue habiendo dificultades para cuantificar la información de deformación cuando las geometrías de deformación producen distribuciones de franjas complejas, altas densidades de franjas y sobre modulación del color.

En su artículo, Briñez et al., 2018 [29] utilizan un conjunto de imágenes sintéticas dentro de una técnica computacional de cambio de fase híbrida aplicada a la fotoelasticidad digital.

En el año 2019, Briñez et al., [32] se presenta un análisis computacional de los efectos Bayer y demodulación en los estudios de fotoelasticidad. Esta investigación demostró que los algoritmos de demodulación introducen errores al registrar imágenes de fotoelasticidad y, en consecuencia, en la evaluación del esfuerzo. La precisión de la información evaluada puede variar en función de los experimentos de fotoelasticidad, como la fuente de luz, la configuración del polariscopio, el modelo de esfuerzo, las aplicaciones de carga, la resolución de la imagen y los algoritmos de demodulación.

En el año 2020, estos autores proponen una metodología de generación de imágenes analíticas por medio de una técnica de load-stepping (escalonamiento de carga) y diferentes fuentes de iluminación. (Briñez et al., 2020 [25]).

Posteriormente se publica la colección de imágenes “Isochromatic-Art” (Briñez et al., 2020 [33]), una colección de más de 100.000 imágenes isocromáticas (estado de polarización único) y sus correspondientes imágenes en escala de grises de mapas de esfuerzo de 224 x 224 píxeles para modelos digitales clásicos de fotoelasticidad.

Para el año 2022, los mismos autores proponen una estrategia de refinamiento para generar una mayor variedad de imágenes analíticas al normalizar, re-escalar y realizar particiones de cada una de las imágenes analíticas generadas, permitiendo así una colección de mayor tamaño y más versátil ante la necesidad de predicción de imágenes reales. (Briñez et al., 2022) [2]

Por otro lado, se han utilizado algunas otras representaciones para analizar la polarización de la luz, como las imágenes térmicas basadas en polarimetría (Gurton et al., 2014) [34] o las imágenes de microscopía de polarización para aplicaciones biomédicas (Chang et al., 2016) [35]. Algunos autores incluso han aplicado modelos de aprendizaje profundo a varios sistemas de imágenes polarimétricas, como la diferencia de polarización, el vector de Stokes y la matriz de Mueller (Li et al., 2022) [36].

Adicional a esto, se encontraron principalmente resultados que involucran estudios generales con generación y análisis de imágenes polarimétricas, como es el caso de Gao et al., 2019 [18], en el que se utilizan imágenes de intensidad con varias orientaciones de polarización y estructuras de fusión de características basadas en CNN.

Es de gran importancia también destacar dentro de la literatura, a algunos autores que han desarrollado las ecuaciones que rigen la distribución de esfuerzos en geometrías típicas de fotoelasticidad digital, para cargas homogéneas y heterogéneas. Dentro de los modelos más comunes se destacan el disco y el anillo bajo compresión diametral, la placa infinita cargada con un orificio circular en el centro, la ménsula curva cargada en extremo libre y la viga vertical bajo presión hidrostática. Todos estos modelos son explicados y desarrollados por Timoshenko et al., 1951 [37]. Adicionalmente, también se trabaja con el modelo de placa con muesca lateral en V, desarrollado por Kramer et al., 2009 [38].

Finalmente, existen también modelos para cargas heterogéneamente distribuidas, como podría ser el caso de los estudios reportados por Tang et al., 2021 [39], donde se realiza el comparativo entre aplicar una carga puntualmente sobre una geometría de disco y distribuir la carga sobre una superficie distinta a un punto único; o también el análisis de un disco sometido a cargas arbitrarias auto equilibradas desarrollado por Ramesh et al., 2022. [40]

## 2.3 Procesamiento de patrones de franjas mediante DCNNs

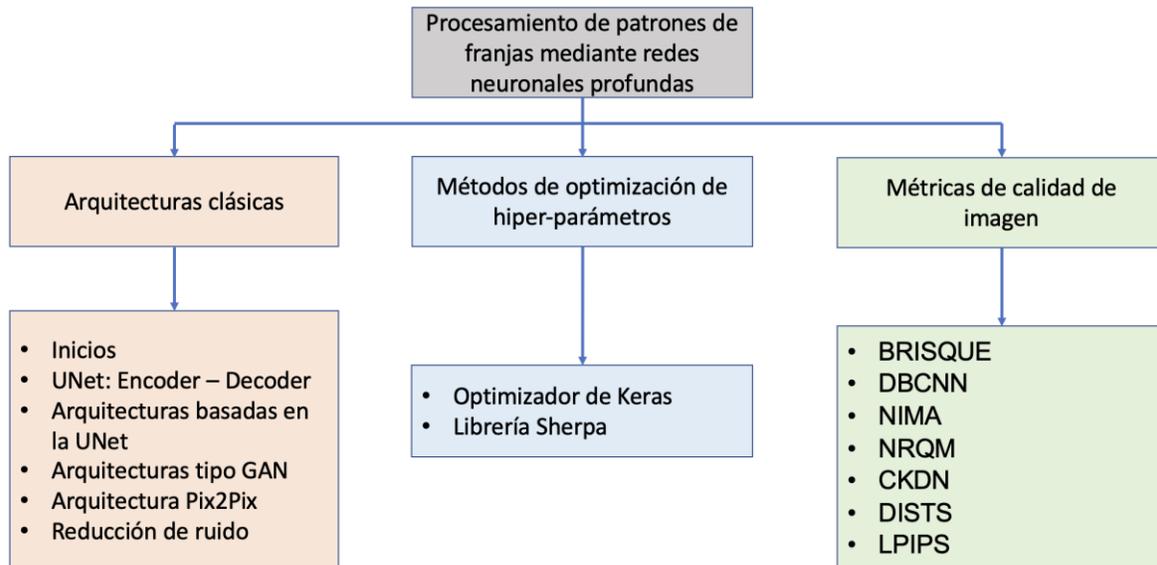
La última temática de búsqueda de información y antecedentes responde especialmente a los objetivos específicos 2 y 3, correspondientes a la elaboración de un método para determinar los hiper-parámetros de las arquitecturas de red neuronal a desarrollar y a su respectiva evaluación, utilizando métricas de calidad de imagen.

Por ello, resulta fundamental realizar una búsqueda de arquitecturas típicas de red neuronal que hayan sido utilizadas en problemas con imágenes de patrones de franjas, desenvolvimiento de fase, regresión de imágenes, entre otros.

Adicional a esto, resulta de bastante importancia consultar fuentes de información donde se haya investigado la optimización de hiper parámetros en algoritmos de aprendizaje de máquinas (Machine Learning ML) o bien, aprendizaje profundo (Deep Learning DL). Finalmente, se realiza una búsqueda de algunas métricas de calidad de imagen clásicas y

otras más modernas, que han sido utilizadas en los entrenamientos y validaciones de este tipo de algoritmos.

La imagen 4 resume los hallazgos dentro de esta temática:



**Imagen 4.** Antecedentes procesamiento de patrones de franjas mediante redes neuronales profundas.

Dentro de las arquitecturas de redes neuronales aplicadas al procesamiento de imágenes con patrones de franjas, son varios los autores que han implementado distintas propuestas a lo largo de los últimos años.

El primer artículo encontrado en trabajar patrones de franjas mediante redes neuronales es el presentado por Dubey et al., 2007 [6], en el que se plantea una arquitectura de red neuronal simple de tipo *Fully Connected*. La salida del modelo es un único valor de tipo entero correspondiente a la fuerza que es aplicada en una geometría, por lo que este estudio podría ser tratado como un problema de regresión. Se utiliza MSE como función de pérdida y se validan los resultados mediante técnicas e imágenes de elementos finitos.

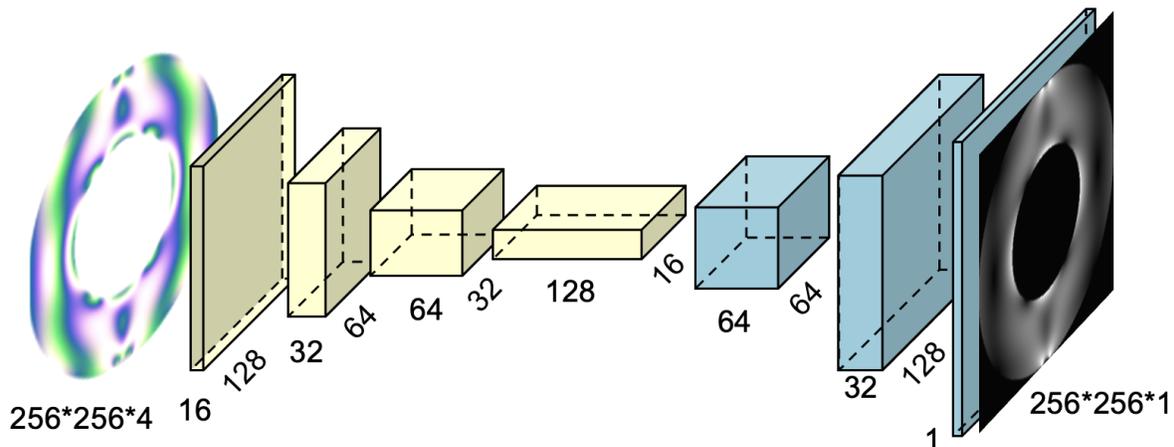
Es de gran importancia también mencionar que las arquitecturas más comunes encontradas para este tipo de problemáticas son las arquitecturas tipo encoder – decoder, también conocidas como arquitectura UNet. Este tipo de redes se utilizan ampliamente en tareas de segmentación y regresión de imágenes debido a su capacidad para captar características relevantes a diferentes escalas. La arquitectura UNet consta de dos partes principales: un "codificador" y un "decodificador". El codificador consiste en una serie de capas de convolución y submuestreo que capturan progresivamente características de nivel superior mediante operaciones de filtrado convolucional y pooling, etapa en la cual se reduce la resolución espacial de la imagen, pero se aumenta la capacidad de extraer características.

Por otro lado, el decodificador utiliza capas de convolución y deconvolución para aumentar la resolución espacial y reconstruir una máscara de segmentación precisa o una imagen de

regresión. Se presentan también conexiones de salto entre el codificador y el decodificador que permiten pasar información de bajo nivel directamente al decodificador, lo que ayuda a fusionar características a diferentes escalas y a mejorar la precisión de la segmentación/regresión.

En resumen, las redes neuronales convolucionales con arquitectura UNet utilizan características aprendidas a distintos niveles de abstracción para lograr una segmentación/regresión de imágenes precisa. La arquitectura de red UNet fue propuesta en 2015 por Ronnenberg et al. [41]

La imagen 5 ilustra la estructura general de una UNet para el desenvolvimiento de fase en un modelo tipo anillo bajo compresión diametral:



**Imagen 5.** Estructura general de una UNet para el desenvolvimiento de fase en un modelo tipo anillo bajo compresión diametral.

Existe una gran variedad de arquitecturas basadas en la UNet encontradas en la literatura, con diferentes configuraciones, hiper-parámetros y finalidades. Se destacan las utilizadas en los estudios realizados por Wang et al., 2019 [8], para desenvolver con éxito campos de fase de osteoblastos de ratón vivos y de la llama dinámica de una vela, o los de Perera et al., 2020 [14] para desenvolver la fase utilizando capas con memoria a corto plazo cuatridireccional espacial SQD-LSTM en la fase intermedia de la red.

También se encontraron algunas redes (con esquema UNet) que realizan específicamente la extracción de campos de esfuerzos en imágenes isocromáticas de fotoelasticidad. Dentro de estas arquitecturas se destacan:

La StressNet (Briñez et al., 2020) [3], una red capaz de demodular los patrones de franjas en una superficie de esfuerzos continua con un buen rendimiento obtenido para una amplia variedad de condiciones experimentales, pero también una red donde queda latente una profundización en las diferencias entre el esfuerzo máximo recuperado con respecto a la información esperada y el efecto de las superficies redondeadas, con el fin de mejorar el proceso de cuantificación del esfuerzo.

La PhotoelastNet (Briñez et al., 2022) [2], proporciona un marco de aprendizaje profundo que permite rápidamente la reconstrucción de fase sin la necesidad de complicados

montajes ópticos. Además, por el hecho de trabajar exclusivamente con patrones isocromáticos, este método evita problemas relacionados con inconsistencias, ambigüedades o franjas isoclínicas. Es un modelo validado en imágenes isocromáticas sintéticas generadas computacionalmente. La PhotoelastNet corresponde a una red mucho más ligera que la StressNet, capaz de alcanzar métricas de calidad de imagen similares.

La red de recuperación de campos de esfuerzos (Tao et al., 2022) [4], una red neuronal de tipo VGG en la cual se optimiza la función de pérdida propuesta en el desarrollo de la PhotoelastNet, por medio de una función de optimización objetiva ponderada con múltiples funciones de pérdida para recuperar la distribución del campo de esfuerzo a partir de patrones de franjas foto elásticas a color.

La StressUnet (Zhao et al., 2022) [42], una red neuronal convolucional basada en U-Net que toma Resnet como columna vertebral. Añade un módulo sencillo denominado *constraint*, que hace que el mapa de esfuerzos sea coherente con la ley física. Acelera la evaluación del esfuerzo y hace realista la detección instantánea empleando el modelo en el procesamiento de datos. Sin embargo, sufre una degradación del rendimiento en el caso de patrones complejos y grandes magnitudes de esfuerzo.

La VDE-Net (Zhao et al., 2022) [43], propone una nueva DCNN con un mecanismo de atención de salto de borde ponderado, a saber, VDE-Net, para resolver el problema de desdoblamiento de fase.

En general, son muchos los autores que han involucrado arquitecturas tipo UNET dentro de la predicción de campos de esfuerzo en imágenes isocromáticas de fotoelasticidad. Existen incluso algunos artículos tipo *review* donde se comparan algunas de estas arquitecturas, tales como los presentados por Wang et al., 2022 [44] o por Zuo et al., 2022 [45]

Se reportan también otras arquitecturas alternativas, diferentes a la UNet, para el problema de procesamiento de imágenes con patrones de franjas. Dentro de estas, se destacan la arquitectura de red neuronal adversaria generativa (GAN) para procesos de recuperación de campos de esfuerzo a partir de imágenes de fotoelasticidad propuesta por Briñez et al., 2020 [12] o también, la arquitectura DeepLabV3+ propuesta por Zhang et al., 2019 [9], la cual involucra un primer proceso de segmentación semántica sobre el mapa de fase envuelto y posteriormente, combina el mapa de fase envuelto original con el resultado de la segmentación para generar fase desdoblada.

De igual manera, se encontró también una implementación de Park et al., 2022 [46] con una arquitectura GAN + Pix2Pix, en la que la arquitectura del generador, que es similar a una UNet, recupera una imagen de fase desdoblada a partir de imágenes de fase envuelta, mientras que el discriminador compara imágenes falsas y reales con capas de convolución. Los resultados mostraron que el modelo propuesto puede reconstruir automáticamente la imagen de fase desdoblada a partir de la imagen de fase envuelta, independientemente de la distancia de reconstrucción.

Finalmente, dentro de la taxonomía de arquitecturas, es también importante mencionar algunos hallazgos en redes neuronales con un enfoque en la reducción de ruido en imágenes con patrones de franjas, tal como lo exponen Yan et al., 2018 [7] o también en procesos de pre-filtrado en imágenes de este tipo, como lo exponen Lin et al., 2019 [11].

En cuanto al tema de métodos de optimización de hiper-parámetros, la bibliografía consultada permitió conocer dos alternativas bastante refinadas para realizar este proceso: El optimizador de Keras y la librería Sherpa:

El optimizador de Keras es una parte fundamental del marco de trabajo de aprendizaje profundo Keras, que se utiliza para ajustar los parámetros de un modelo de manera iterativa durante el proceso de entrenamiento. El objetivo principal del optimizador es minimizar una función de pérdida o maximizar una función de rendimiento, lo que implica ajustar los pesos y sesgos del modelo para mejorar su capacidad de generalización y hacer predicciones más precisas. Keras proporciona una variedad de optimizadores populares, como Adam, RMSprop y Gradiente Descendente Estocástico (SGD), cada uno con sus propias características y algoritmos de optimización. Estos optimizadores permiten a los usuarios ajustar y personalizar los hiper-parámetros según las necesidades específicas del modelo y del conjunto de datos en cuestión. En su artículo, Mohamad et al., 2021 [47] utiliza este optimizador en un problema de Deep Learning con los datasets públicos MNIST y CIFAR-10 y lo compara con la librería AiSara tuner.

Por su parte, la librería Sherpa fue desplegada en el año 2020 por Hertel et al. [48] Sherpa es un software flexible de código abierto para la optimización de hiper-parámetros de modelos de aprendizaje automático. Ofrece al usuario varios algoritmos de optimización de hiper-parámetros intercambiables, cada uno de los cuales puede ser útil en distintas fases del desarrollo del modelo. Su panel de control interactivo permite al usuario supervisar y analizar los resultados de múltiples optimizaciones de hiper-parámetros en tiempo real. También permite al usuario ver patrones en el rendimiento de los hiper-parámetros para juzgar la solidez de los ajustes individuales. Sherpa puede utilizarse en un ordenador portátil o de forma distribuida en un clúster. En resumen, en lugar de una caja negra que escupe una configuración de hiper-parámetros, Sherpa proporciona las herramientas que un investigador necesita cuando realiza la exploración y optimización de hiper-parámetros para el desarrollo de modelos de aprendizaje automático.

Por último, para la temática de métricas de calidad de imagen, comprendida netamente en el objetivo específico 3, se realizó una investigación que permitiera complementar las métricas MSE, SSIM y PSNR, que son las clásicamente utilizadas en este tipo de estudios.

Dentro de las métricas encontradas, es posible encontrar dos categorías: Métricas “No reference” (NR) y métricas “Full Reference” (FR). Como su nombre lo indica, las métricas NR son aquellas capaces de otorgar una calificación a una imagen sin requerir de otra imagen de referencia con la cual comparar. Estas métricas generalmente se generan mediante algoritmos pre-entrenados que se basan en factores y percepciones humanas para brindar una valoración a una imagen teniendo en cuenta sus características. Es el caso de las métricas BRISQUE, DBCNN, NIMA, NRQM, CKDN, entre otras.

Por su parte, las métricas FR corresponden a aquellas en las que se cuenta con una imagen de referencia con la cual comparar una imagen obtenida por una red neuronal. Es el caso de las métricas DISTS y LPIPS, entre otras.

A continuación, se explica brevemente el funcionamiento de cada una de estas métricas.

BRISQUE (*Blind Referenceless Image Spatial Quality Evaluator*) (Mittal et al., 2012 [49]): Es un algoritmo de evaluación de calidad sin referencia (NR) basado en estadísticas de

escenas naturales, que opera en el dominio espacial. Compara la imagen dada con un modelo por defecto calculado a partir de imágenes de escenas naturales con distorsiones similares. Una puntuación menor indica una mejor calidad perceptiva.

DBCNN (Blind Image Quality Assessment Using A *Deep Bilinear Convolutional Neural Network*) (Zhang et al., 2020 [50]): Consiste de un modelo bilineal profundo para la evaluación de imágenes que maneja tanto distorsiones sintéticas como auténticas en dos redes neuronales convolucionales (CNN) cada una de las cuales se especializa en un escenario de distorsión. Para las distorsiones sintéticas, se pre-entrena una CNN para para clasificar el tipo y el nivel de distorsión de la imagen, para lo cual se dispone de datos de entrenamiento a gran escala. Para las distorsiones auténticas, se adopta una CNN pre-entrenada para la clasificación de imágenes.

NIMA (*Neural Image Assessment*) (Talebi et al., 2017 [51]): Predice la distribución de las puntuaciones de las opiniones humanas utilizando una red neuronal convolucional. La arquitectura tiene la ventaja de ser significativamente más simple que otros métodos con un rendimiento comparable. La red resultante puede utilizarse no sólo para puntuar imágenes de forma fiable y con una alta correlación con la percepción humana, sino también para ayudar a la adaptación y optimización de los algoritmos de edición/mejora de fotografías en un proceso fotográfico.

NRQM (*No-Reference Quality Metric for Single-Image Super-Resolution*) (Chao et al., 2017 [52]): Consiste en una métrica sin referencia aprendida a partir de las puntuaciones perceptivas visuales. Consta de un modelo de regresión de dos etapas para predecir las puntuaciones de calidad de las imágenes de superresolución sin referirse a las imágenes reales. Los amplios resultados experimentales muestran que es eficaz y eficiente para evaluar la calidad de las imágenes de superresolución basándose en la percepción humana.

CKDN (*Conditional Knowledge Distillation Network*) (Zheng et al., 2021 [53]): Este modelo extrae la información de referencia de las imágenes degradadas. La destilación se consigue mediante el aprendizaje de un espacio de referencia, en el que se fomenta que varias imágenes degradadas compartan las mismas estadísticas de características. Este trabajo proporciona una métrica potente y diferenciable para las restauraciones de imágenes ciegas, especialmente para los métodos basados en GAN.

DISTS (*Deep Image Structure and Texture Similarity*) (Ding et al., 2022 [54]): Utiliza una red neuronal convolucional, se construye una función inyectiva y diferenciable que transforma las imágenes en representaciones multi-escala sobre-completas. Los promedios espaciales de los mapas de características en esta representación capturan la apariencia de la textura, en el sentido de que proporcionan un conjunto de restricciones estadísticas suficientes para sintetizar una amplia variedad de patrones de textura. Resulta un método de calidad de imagen que combina las correlaciones de estos promedios espaciales ("similitud de textura") con las correlaciones de los mapas de características ("similitud de estructura").

LPIPS (*Learned Perceptual Image Patch Similarity*) (Zhang et al., 2018 [55]): Recientemente, la comunidad de aprendizaje profundo ha descubierto que las características de la red VGG entrenadas en la clasificación de ImageNet han sido notablemente útiles como pérdida de entrenamiento para la síntesis de imágenes. Esta

métrica propone el uso de características profundas y supera a todas las métricas anteriores con un amplio margen. Y lo que es más sorprendente, este resultado no se limita a las características VGG entrenadas en ImageNet, sino que se mantiene en diferentes arquitecturas profundas y niveles de supervisión (supervisada, autosupervisada o incluso no supervisada).

### 3. Marco Teórico

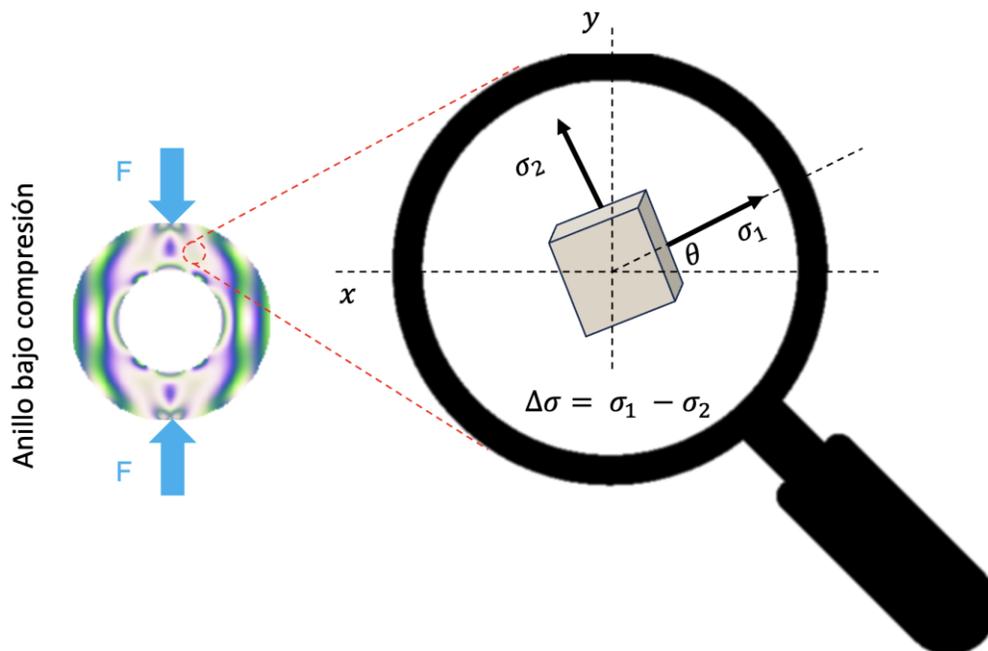
En este capítulo se presentan y definen algunos conceptos que ayudan a establecer un contexto más amplio, para el entendimiento del problema planteado: Fotoelasticidad, polariscopio, extracción de esfuerzos en imágenes de fotoelasticidad, cámaras e imágenes multipolarizadas, métricas de calidad de imagen, entrenamiento y optimización de red neuronal.

#### 3.1 Fotoelasticidad

La fotoelasticidad es un método experimental para el análisis de esfuerzos y deformaciones. Se basa en los fenómenos experimentados por las ondas electromagnéticas a su paso por materiales translúcidos, en particular la polarización de la luz, que ocurre como consecuencia de los esfuerzos presentes en los cuerpos sometidos a esfuerzos.

Este método es útil para el estudio de objetos de forma irregular y esfuerzos localizados, como piezas de maquinaria con muescas, curvas, ranuras, agujeros y/o hendiduras. Adicionalmente, proporciona un medio visual para observar las características de los esfuerzos por medio de patrones de franjas proyectados sobre una pantalla o sobre una película fotográfica.

La imagen 6 ilustra el estado de esfuerzos principales en un elemento infinitesimal de una geometría de anillo bajo carga a compresión. Es posible observar el patrón de franjas generado en la geometría luego de aplicar una carga y de disponer un montaje tipo polariscopio.



**Imagen 6.** Representación gráfica de los componentes principales de los esfuerzos. Adaptado de (Briñez et al, 2020) [2]

Por simple observación podrían determinarse las regiones en que se concentran los esfuerzos, que en este caso coinciden con los puntos de aplicación de la carga y los bordes del agujero central, quien ocasiona una concentración de esfuerzos.

Sin embargo, se necesitaría una técnica más profunda para un análisis de mayor precisión sobre la tensión, compresión, fuerzas de cizalladura (cortante) y deformaciones en cualquier punto de un objeto con esfuerzos triaxiales, o bien, para la cuantificación de las magnitudes de las diferencias de esfuerzos principales presentes en la pieza cargada. La inspección visual únicamente permite conocer los puntos máximos de concentración de esfuerzos, más no su magnitud.

Por medio de la fotoelasticidad es posible realizar los siguientes procedimientos:

- Identificación de áreas críticas (regiones de concentración de esfuerzo).
- Distribución de esfuerzos alrededor de concentradores como agujeros, muescas, entallas y sitios potenciales a posibles fallos.
- Pruebas de esfuerzo variable, sin necesidad de desmontar o reemplazar el material.
- Cuantificación de esfuerzos residuales.

La fotoelasticidad ha tenido aplicaciones exitosas en varios campos de manufactura y construcción donde es empleada. Se destacan los usos en la industria automotriz, maquinaria agrícola, aviación, construcciones civiles, ingeniería, construcción naval, equipos de oficina, puentes, diseño de accesorios, herramientas, y muchas otras aplicaciones.

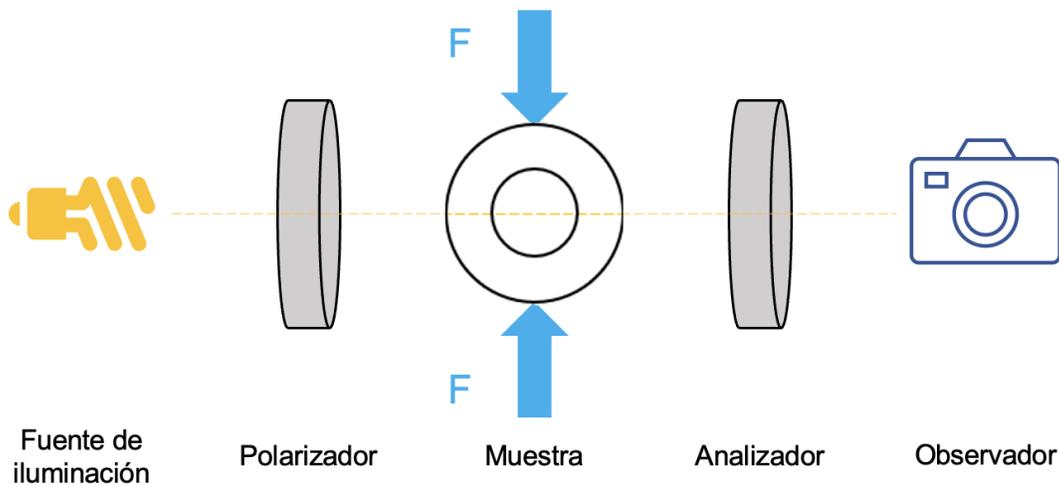
Los avances tecnológicos a nivel de computación y las técnicas del procesamiento digital de imágenes han hecho de la fotoelasticidad más eficiente y confiable para la solución de problemas de ingeniería. (Brieva et al., 2011) [56]

### 3.2 Polariscopeo

El polariscopio es un instrumento óptico, que en su operación utiliza las propiedades de la luz polarizada. El polariscopio se compone esencialmente de una fuente de iluminación que suele ser monocromática (más no se limita únicamente a ella), polarizadores (analizadores), retardadores de cuarto de onda, el elemento bajo carga (muestra) y una cámara (observador) quien realiza la captura de las franjas presentes en la pieza como resultado del retardo de la luz que atraviesa al elemento.

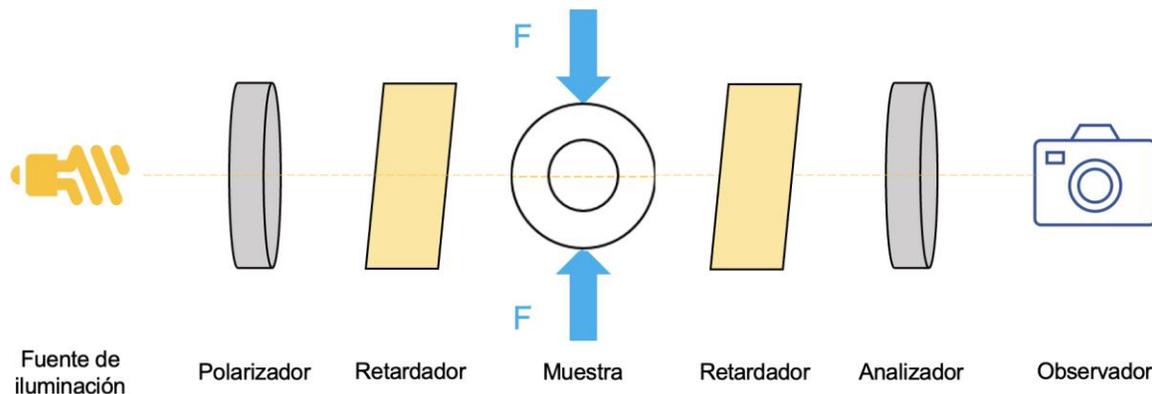
En el análisis experimental de esfuerzos, se utilizan dos tipos de polariscopios, el plano y el circular. Los nombres provienen del tipo de luz polarizada que utilizan para su operación. El polariscopio lineal, es un instrumento que permite observar y medir el estado de polarización de objetos translúcidos. Está compuesto de un polarizador lineal y de un analizador. El polarizador se encuentra fijo y es el que define la dirección de polarización de la luz y el segundo, orientable, se utiliza para observar la rotación de la polarización. El objeto de interés se pone entre ambos. (Brieva et al., 2011) [56].

La imagen 7 ilustra el esquema de un polariscopio plano (lineal):



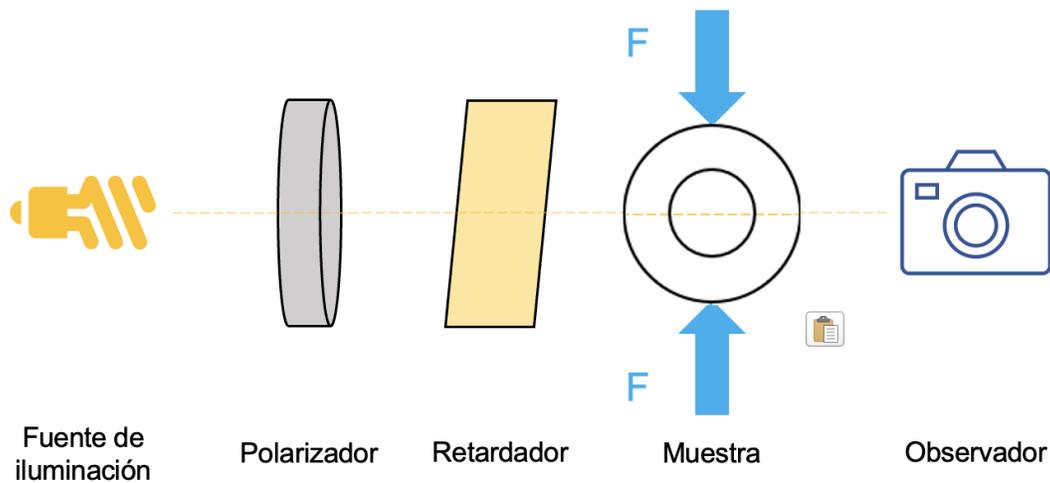
**Imagen 7.** Esquema de un montaje de polariscopio plano.

Por su parte, el polariscopio circular se compone de los mismos elementos que uno lineal, pero con la adición de elementos retardadores de fase, que, como su nombre lo indica, generan un retardo en la onda de luz que atraviesa al montaje. La imagen 8 ilustra el esquema de un polariscopio circular:



**Imagen 8.** Esquema de un montaje de polariscopio circular.

En el caso específico de utilizar una cámara multipolarizada, la cual incluye dentro de sí a un polarizador, sería redundante contar con dos polarizadores dentro del montaje del polariscopio. En este caso específico suele eliminarse este segundo polarizador (también denominado analizador) y, en ocasiones, también se elimina uno de los dos retardadores de cuarto de onda. Este tipo de montaje recibe el nombre de polariscopio semicircular. La imagen 9 ilustra el esquema de un polariscopio semicircular:



**Imagen 9.** Esquema de un montaje de polariscopio semicircular.

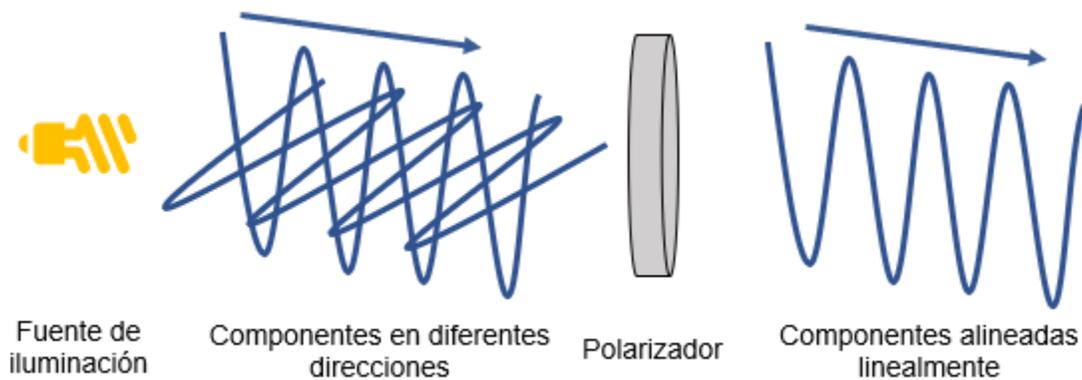
### 3.3 Extracción de esfuerzos en imágenes de fotoelasticidad

La relación existente entre los patrones de franjas isocromáticas y la distribución de esfuerzos en una pieza de un material birrefringente se origina de la siguiente manera:

En un montaje de polarización, la fuente de iluminación emite rayos de luz no polarizada. Las componentes de campo eléctrico ' $E$ ' de la luz pueden ser representadas a través de la ecuación 1, donde ' $A$ ' es la amplitud de la onda, ' $t$ ' el tiempo y ' $\omega$ ' la frecuencia angular.

$$E = Ae^{i\omega t} \quad (1)$$

Por su parte, la imagen 10 ilustra la generación de haces de luces no polarizadas desde una fuente de iluminación y su resultado linealmente alineado luego de atravesar por un polarizador:



**Imagen 10.** Representación de la polarización lineal de la luz.

Al atravesar un medio retardador, se introduce un retardo relativo en la componente de campo eléctrico, tal retardo se expresa en la ecuación 2:

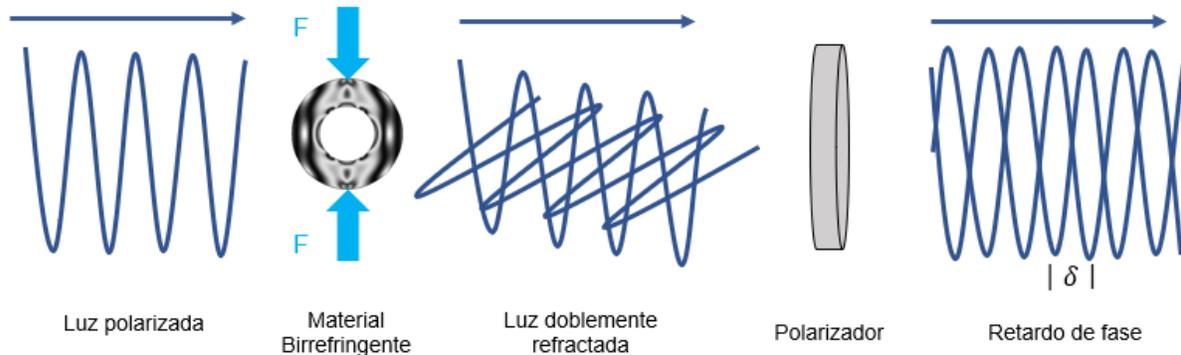
$$\Delta = \frac{2\pi\delta}{\lambda} \quad (2)$$

Donde ' $\Delta$ ' representa el cambio de fase relativo, en función del retardo de la onda ' $\delta$ ' que se genera por la diferencia de los índices de refracción de los medios donde viaja la onda y ' $\lambda$ ' es la longitud de onda.

Cuando un rayo de luz polarizada pasa a través de un material birrefringente emergen de él dos componentes linealmente polarizadas, perpendiculares entre sí, que al alinearlas nuevamente utilizando un segundo instrumento polarizador crean un retardo de fase ' $\delta$ ' entre ellas, que está asociado al espesor del material ' $h$ ' y sus índices de refracción ' $n$ ' como se expresa en la ecuación 3:

$$\delta = h(n_1 - n_2) \quad (3)$$

La imagen 11 presenta un esquema del comportamiento del rayo de luz polarizado al entrar al material birrefringente, cuando sale de él y cuando sus componentes polarizadas son alineadas nuevamente en el segundo polarizador, creando entre ellas un retardo de fase que da origen a la formación de las franjas de color asociadas al espesor, a los índices de refracción y a los esfuerzos aplicados sobre el material:



**Imagen 11.** Representación de la luz a través de un material birrefringente.

Existen materiales que experimentan birrefringencia al ser sometidos a esfuerzos y deformaciones mecánicas, que al ser analizados con la ley de tensión óptica se logra realizar una asociación entre los esfuerzos del material ' $\sigma$ ', con los índices de refracción ' $n_1$ ' y ' $n_2$ ' y el coeficiente de esfuerzo óptico ' $C$ ', y por lo tanto reconstruir la ecuación 3 en términos de la diferencia de esfuerzos principales, tal como se presenta en la ecuación 4:

$$\delta = hC(\sigma_1 - \sigma_2) \quad (4)$$

Del análisis de esta ecuación, se puede inferir que manteniendo constantes los parámetros del experimento (espesor ' $h$ ' y carga), se pueden describir los esfuerzos principales con la medición del retardo de fase y el conteo de las franjas; por lo que, en su mayoría, los

trabajos en fotoelasticidad se ocupan de las técnicas para la medición y el cálculo de los parámetros antes mencionados.

Al despejar las direcciones de esfuerzos principales en función del retardo de fase, la longitud de onda, el coeficiente de esfuerzo óptico y el espesor del material se obtiene la ecuación 5, que permite asignar como orden de franjas 'N' a  $\frac{\Delta}{2\pi}$  y como valor de la franja  $\frac{\lambda}{c}$  a ' $\sigma_f$ '. Tales expresiones permiten identificar los esfuerzos principales a partir del retardo de fase y del conteo de las franjas. (Briñez et al., 2013) [1]

$$(\sigma_1 - \sigma_2) = \frac{\Delta\lambda}{2\pi Ch} \quad (5)$$

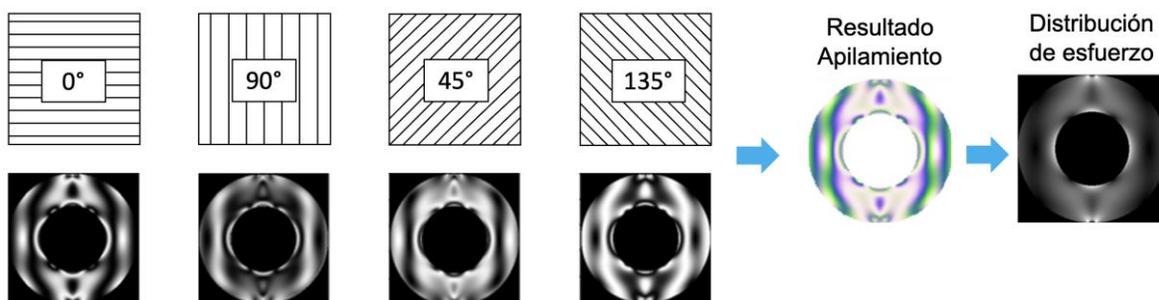
O, equivalentemente:

$$\delta = \frac{2\pi Ch(\sigma_1 - \sigma_2)}{\lambda} \quad (6)$$

### 3.4 Cámaras e imágenes multipolarizadas

Este tipo de cámaras es de gran importancia en el planteamiento y desarrollo de esta tesis, al ser un factor diferenciador dentro de las problemáticas estudiadas hasta el momento.

Estas cámaras incorporan un polarizador capaz de obtener hasta cuatro estados de polarización de una misma imagen dentro de una misma captura: a 0°, 45°, 90° y 135°, respectivamente, lo que permite obtener un estado más amplio de la condición bajo la cual se encuentra la geometría estudiada. La imagen 12 ilustra el resultado de apilar cuatro imágenes monocromáticas, en diferentes estados de polarización, para generar una imagen multipolarizada.



**Imagen 12.** Apilamiento de cuatro imágenes monocromáticas en diferentes estados de polarización para la generación de una imagen multipolarizada.

### 3.5 Métricas de calidad de imagen

Tal como su nombre lo indica, las métricas de calidad de imagen permiten realizar una evaluación cuantitativa de los valores presentes en una imagen, o bien, de la semejanza

entre dos imágenes distintas. Para ello, las métricas toman los valores contenidos en los píxeles de las imágenes y realizan un cómputo específico en el que evalúan o comparan dicho(s) valor(es) con algunos parámetros predefinidos, o con los valores de otra imagen que sirve de comparación, según sea el caso.

En general, es posible encontrar tres métricas de calidad de imagen con bastante frecuencia en estudios que involucran el análisis y el procesamiento de imágenes: MSE, PSNR y SSIM, sin limitarse únicamente en estas, tal como se pudo evidenciar con otras métricas encontradas en la construcción del estado del arte. Todo esto, con el fin de llevar una cuantificación numérica de las imágenes obtenidas por los métodos de aprendizaje profundo, en comparación con las imágenes objetivo.

A continuación, se detallan las tres métricas de uso más frecuente:

### 3.5.1 Error cuadrático medio, MSE

Es el promedio de la sumatoria de la diferencia de cada par de valores de los píxeles en cada imagen, uno a uno, elevado al cuadrado. La ecuación 7 define al MSE:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_i)^2 \quad (7)$$

### 3.5.2 Proporción Máxima de Señal a Ruido, PSNR

Consiste en la multiplicación de 10 por el logaritmo en base 10 del máximo valor que puede tomar un píxel en la imagen objetivo (al cuadrado) dividido por el MSE. La ecuación 8 define al PSNR:

$$PSNR = 10 \cdot \log_{10} \left( \frac{MAX_I^2}{MSE} \right) \quad (8)$$

### 3.5.3 Medida del índice de similitud estructural, SSIM

Esta métrica supone la creación de dos ventanas que recorren las imágenes a comparar. Una de las ventanas lo hace en sentido horizontal, a la cual se le denomina ventana en x y la otra lo hace en sentido vertical, la denominada ventana en y. La ecuación 9 define al SSIM:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x+\sigma_y+c_2)} \quad (9)$$

Donde:

$\mu_x$  corresponde al valor medio de los píxeles recorridos por la ventana en x,  
 $\mu_y$  corresponde al valor medio de los píxeles recorridos por la ventana en y,  
 $\sigma_x$  corresponde a la varianza de los píxeles recorridos por la ventana en x,  
 $\sigma_y$  corresponde a la varianza de los píxeles recorridos por la ventana en y,  
 $\sigma_{xy}$  corresponde a la covarianza de los píxeles recorridos por ambas ventanas,

$c_1, c_2$  corresponden a dos constantes para estabilizar la división y dependen del rango de valores presente en los píxeles de las imágenes.

### 3.6 Entrenamiento y optimización de redes neuronales

El entrenamiento y optimización de redes neuronales es un área fundamental en el campo de la inteligencia artificial y el aprendizaje automático. Las redes neuronales son modelos computacionales inspirados en el funcionamiento del cerebro humano, compuestas por capas de neuronas interconectadas. El proceso de entrenamiento de una red neuronal consiste en ajustar los pesos y los sesgos de las conexiones entre las neuronas para que la red pueda aprender a realizar tareas específicas, como reconocimiento y regresión de imágenes, procesamiento de lenguaje natural, regresión y clasificación de valores, entre otros.

La optimización de las redes neuronales busca mejorar su desempeño mediante técnicas que buscan minimizar una función de pérdida o error, también conocida en múltiples fuentes como función de pérdida. Estas técnicas incluyen algoritmos de optimización, como el SGD: descenso de gradiente, que ajustan los parámetros de la red para reducir gradualmente el error durante el entrenamiento. Además, se utilizan técnicas de regularización para evitar el sobreajuste, que ocurre cuando la red memoriza los datos de entrenamiento y se vuelve incapaz de generalizar adecuadamente frente a datos desconocidos, que no fueron utilizados en el entrenamiento.

Dentro del campo de la optimización, existen diversas técnicas y enfoques, como el uso de diferentes funciones de activación, la aplicación de técnicas de regularización, como la eliminación de neuronas (también conocida como *dropout*), la inicialización adecuada de los pesos de la red, entre otros.

En general, los procesos de optimización se pueden realizar de manera iterativa por medio de variaciones en la arquitectura y en los hiper-parámetros de la red, sea de manera manual o con la ayuda de librerías y softwares que automatizan estos procesos.

En resumen, el entrenamiento y la optimización de redes neuronales son procesos esenciales para lograr un rendimiento óptimo en tareas de inteligencia artificial. El dominio de estas técnicas es crucial para desarrollar modelos de aprendizaje automático eficaces y robustos, capaces de abordar desafíos complejos en diferentes áreas de aplicación.

## 4. Generación de la colección de imágenes

En este capítulo se presenta el proceso de construcción de una colección de imágenes multipolarizadas de fotoelasticidad digital en modelos tradicionales bajo carga, lo cual atiende al objetivo específico I de esta tesis y es fundamental para el entrenamiento de las redes que se desarrollarán.

El capítulo se estructurará de la siguiente manera: en primer lugar, se describirá el procedimiento analítico realizado para la generación de las imágenes multipolarizadas en modelos clásicos de análisis de esfuerzos. A continuación, se detallarán los pasos y procedimientos para la representación de imágenes multipolarizadas en dos formatos: utilizando un método de apilamiento de imágenes y construyendo imágenes a partir del vector Stokes. Luego, se explica un proceso de aumento de datos, realizado con el fin de generar una mayor versatilidad en las predicciones de las redes a entrenar. Finalmente, se expone la creación de algunos modelos analíticos adicionales que son generados netamente con la finalidad de validación y evaluación de las redes ante imágenes desconocidas, pues no serán usadas en los procesos de entrenamiento.

En resumen, este capítulo constituye un paso fundamental en la consecución del primer objetivo específico y del objetivo general de esta tesis, al proporcionar una metodología detallada y rigurosa para la construcción de una colección de imágenes multipolarizadas de fotoelasticidad digital.

### 4.1 Método

Al no contar con amplias colecciones públicas de imágenes multipolarizadas de fotoelasticidad, y teniendo en cuenta la compleja configuración y coste de los polariscopios necesarios para obtener imágenes reales, ha sido necesario avanzar hacia enfoques híbridos en los que las imágenes se derivan de simulaciones computacionales, pero incluyen datos experimentales.

Un proceso similar fue utilizado en la construcción de la colección pública Isochromatic-Art (Briñez et al., 2020 [33]), para la construcción analítica de imágenes isocromáticas de un solo estado de polarización.

Con el objetivo de generar una colección similar a esta, pero con imágenes multipolarizadas, se siguió la metodología utilizada en su construcción y se adaptó con el fin de generarlas, en cuatro estados de polarización:  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  y  $135^\circ$ .

Para alcanzar este objetivo, se comenzó realizando una investigación sobre cámaras multipolarizadas típicas que podían ser utilizadas en un montaje real de polariscopio, para la captura de este tipo de imágenes. Se hacía necesario además, que las respuestas espectrales de los sensores de las cámaras consultadas estuvieran disponibles, con la finalidad de conocerlas y replicarlas en la generación analítica de las imágenes multipolarizadas.

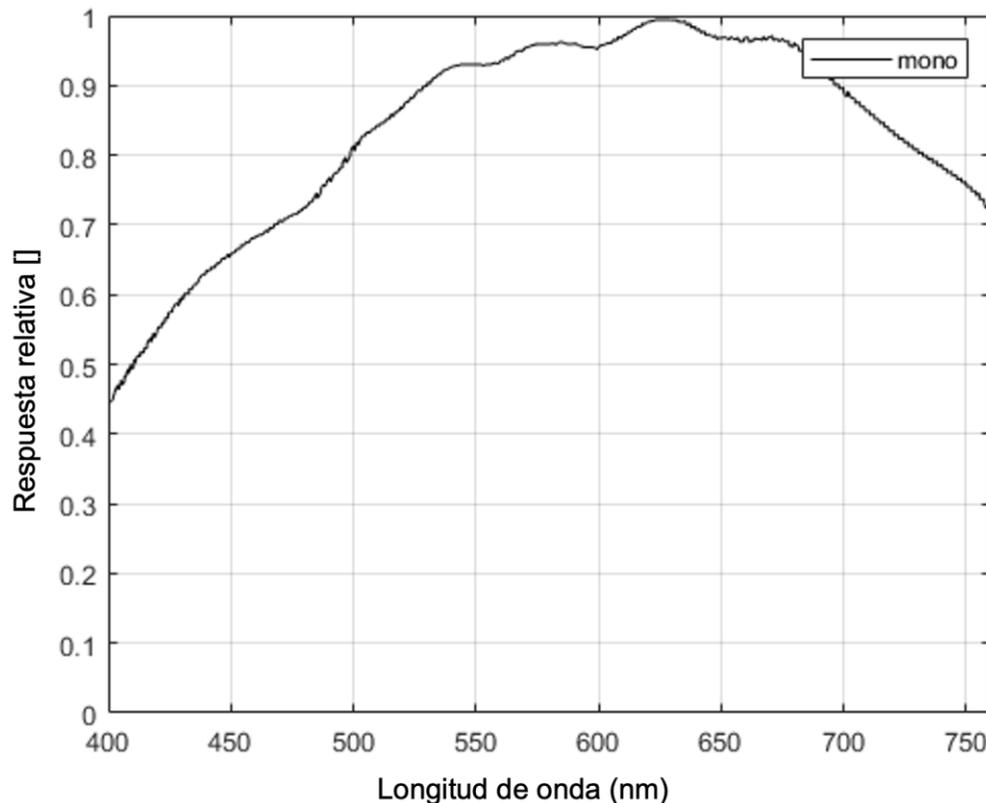
Luego de realizar la consulta, se encontró que algunas cámaras como la BFS-U3-51S5PC-C USB 3.1 Blackfly® S (Edmund Optics) [57] y la VCXU-50MP (Baumer) [58], incorporaban

sensores multipolarizados de última tecnología: Policromático como el Sony IMX250MYR y monocromático como el Sony IMX250MZR, respectivamente.

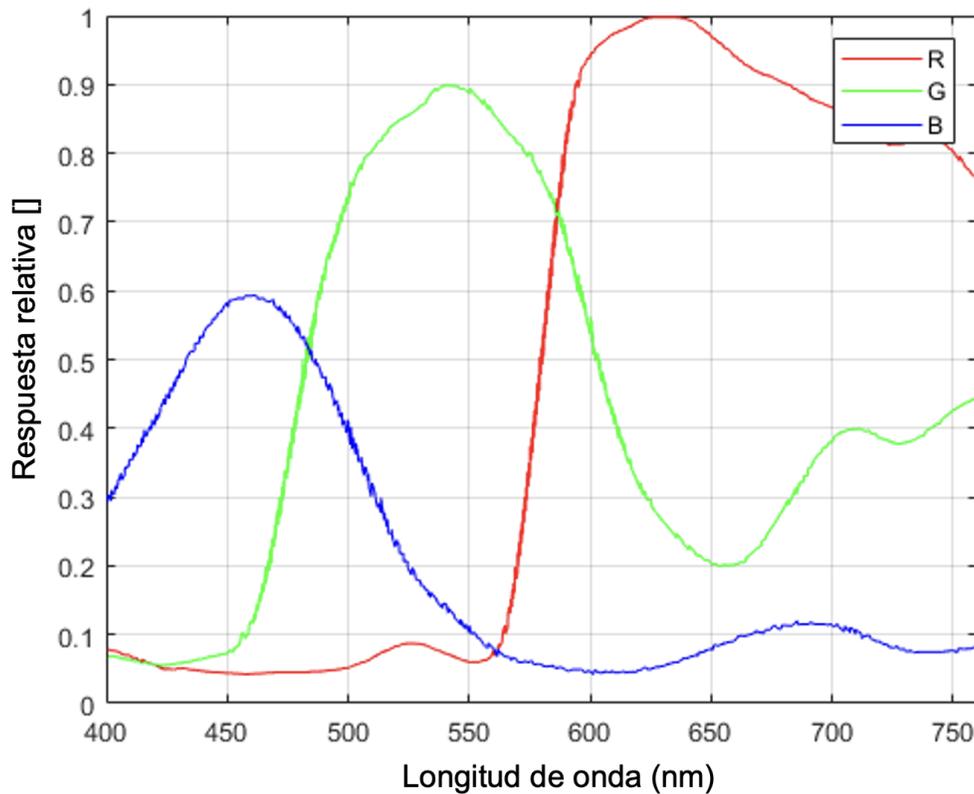
Adicional a esto, algunos fabricantes proporcionan gráficos con la respuesta espectral de estos sensores (IMPERX) [59]. Sin embargo, estos gráficos no son suficientes para extraer la respuesta espectral en su totalidad, pues se requiere conocer un valor más preciso para cada longitud de onda que pueda tomar la fuente de iluminación.

Para lograr este objetivo, se requirió entonces de la ayuda del software de código abierto WebPlotDigitizer (Rohatgi et al., 2019) [60], con el que es posible extraer datos numéricos de gráficos de manera vectorial, para cierto número de pasos definido.

Con ayuda de este software, fue posible realizar la extracción de las respuestas espectrales relativas de estos sensores, ante la presencia de una fuente de iluminación, para diferentes longitudes de onda. Las imágenes 13 y 14 ilustran la aproximación obtenida luego de extraer los valores utilizando el software y partiendo de las respuestas brindadas por el fabricante, para el sensor monocromático y el policromático, respectivamente:



**Imagen 13.** Respuesta espectral relativa vs Longitud de onda para sensor monocromático IMX250MZR. Adaptado de (IMPERX) [59].



**Imagen 14.** Respuesta espectral relativa vs Longitud de onda para sensor policromático IMX250MZYR. Adaptado de (IMPERX) [59].

En la cámara policromática, se consideran todos los canales (R, G, B), mientras que en la cámara monocromática, la respuesta espectral se replica para los tres canales de imagen.

Una vez se tienen las respuestas espectrales de estas cámaras, se deben definir las fuentes de iluminación que serán utilizadas para la simulación, estas son nueve fuentes de luz LED: Concord Beacon Muse LED, Cooper DL11 WS WW LED, Cooper RXD2 RWS WW LED, Fluorescente, Pantalla LCD, Philips EnduraLED MR16 LED, Sylvania UltraLED Par 30LN, Toshiba ECore Par30s 23Deg LED y Willard LEDGO CN 600SC LED. Estas fuentes fueron utilizadas en la construcción de la colección pública Isochromatic-Art (Briñez et al., 2020 [33]) y su comportamiento espectral me fue compartido por los autores.

Posterior a esto, se hace necesario definir algunas propiedades adicionales del material a simular, de las geometrías consideradas y del experimento en sí.

Dentro de los materiales birrefringentes, uno de los más clásicos es el acrílico, también conocido como polimetilmetacrilato o PMMA, por lo que se decide trabajar con este material. El coeficiente óptico de este material es  $3.5 \cdot 10^{-12}$ . Se utilizará un espesor  $H = 0.01 \text{ m}$  (10 mm) en todos los modelos.

En cuanto a los modelos a generar, se recurre a algunas fuentes que han estudiado la distribución de esfuerzos en modelos tradicionales de fotoelasticidad, para extraer de allí las ecuaciones de estos. A continuación se enuncian dichas ecuaciones y la carga máxima

para la que se simuló cada modelo, teniendo en cuenta que se decidió correr un aumento de carga lineal desde 0 N hasta el valor máximo determinado, en un total de 100 pasos de carga:

**Modelo de disco a compresión:** Para este modelo, se toma una carga máxima  $P = 3500 N$  y un radio  $R = 0.05 m$  (5 cm). La distribución de esfuerzos para este modelo fue descrita por Junior et al., 2012 [61] por medio de las ecuaciones 10 a la 12, donde  $\sigma_x^e$ ,  $\sigma_y^e$  y  $\tau_{xy}^e$  corresponden a las componentes del esfuerzo y el superíndice  $e$  indica que los valores analíticos son exactos:

$$\sigma_x^e(x, y) = \frac{-2P}{\pi H} \left\{ \frac{(R-y)x^2}{[x^2+(R-y)^2]^2} + \frac{(R+y)x^2}{[x^2+(R+y)^2]^2} - \frac{1}{2R} \right\} \quad (10)$$

$$\sigma_y^e(x, y) = \frac{-2P}{\pi H} \left\{ \frac{(R-y)^3}{[x^2+(R-y)^2]^2} + \frac{(R+y)^3}{[x^2+(R+y)^2]^2} - \frac{1}{2R} \right\} \quad (11)$$

$$\tau_{xy}^e(x, y) = \frac{2P}{\pi H} \left\{ \frac{(R-y)^2 x}{[x^2+(R-y)^2]^2} + \frac{(R+y)^2 x}{[x^2+(R+y)^2]^2} \right\} \quad (12)$$

**Modelo de anillo a compresión:** Para este modelo, se toma una carga máxima  $P = 3200 N$ , un radio externo  $R2 = 0.05 m$  (5 cm), un radio interno  $R1 = 0.025 m$  (2.5 cm) y un ángulo de contacto  $\Theta$  de  $3^\circ$ . La distribución de esfuerzos para este modelo fue descrita por Tokovyy et al., 2010 [62] por medio de las ecuaciones en coordenadas polares 13 a la 15:

$$\sigma_r(\rho, \varphi) = \frac{p}{\pi} \left[ -\frac{\rho^2 - k^2}{1 - k^2} \frac{2\Theta}{\rho^2} + \sum_{m=1}^{\infty} \frac{(-1)^m \mathcal{R}_m(\rho)}{m} \sin(2m\Theta) \cos(2m\varphi) \right] \quad (13)$$

$$\sigma_\varphi(\rho, \varphi) = \frac{p}{\pi} \left[ -\frac{\rho^2 + k^2}{1 - k^2} \frac{2\Theta}{\rho^2} + \sum_{m=1}^{\infty} \frac{(-1)^m \Phi_m(\rho)}{m} \sin(2m\Theta) \cos(2m\varphi) \right] \quad (14)$$

$$\sigma_{r\varphi}(\rho, \varphi) = \frac{p}{\pi} \sum_{m=1}^{\infty} \frac{(-1)^m \mathcal{S}_m(\rho)}{m} \sin(2m\Theta) \sin(2m\varphi) \quad (15)$$

Donde:

$$p = \frac{P}{2 * R2 * H * \Theta} \quad (16)$$

$$k = \frac{R1}{R2} \quad (17)$$

$$\mathcal{R}_m(\rho) = (2(m+1)\rho^{-2m} - (2m+1)k^2\rho^{-2(1+m)} - k^{2(1-2m)}\rho^{-2(1-m)})A_m - (2(m-1)\rho^{2m} - (2m-1)k^2\rho^{-2(1-m)} + k^{2(1+2m)}\rho^{-2(1+m)})B_m \quad (18)$$

$$\Phi_m(\rho) = \left( (2m+1)k^2\rho^{-2(1+m)} - 2(m-1)\rho^{-2m} + k^{2(1-2m)}\rho^{-2(1-m)} \right) A_m - \left( (2m-1)k^2\rho^{-2(1-m)} - 2(m+1)\rho^{2m} - k^{2(1+2m)}\rho^{-2(1+m)} \right) B_m \quad (19)$$

$$S_m(\rho) = \left( 2m\rho^{-2m} - (2m+1)k^2\rho^{-2(1+m)} + k^{2(1-2m)}\rho^{-2(1-m)} \right) A_m + \left( 2m\rho^{2m} - (2m-1)k^2\rho^{-2(1-m)} - k^{2(1+2m)}\rho^{-2(1+m)} \right) B_m \quad (20)$$

Y las constantes  $A_m$  y  $B_m$  tienen la forma:

$$A_m = \frac{k^{2(1+m)}(k^{2m}-k^{-2m}) - 2m(1-k^2)}{4m^2(1-k^2)^2 - k^2(k^{2m}-k^{-2m})^2} \quad (21)$$

$$B_m = \frac{2m(1-k^2) - k^{2(1-m)}(k^{2m}-k^{-2m})}{4m^2(1-k^2)^2 - k^2(k^{2m}-k^{-2m})^2} \quad (22)$$

**Modelo de placa infinita:** Para este modelo, se toma una carga máxima  $P = 7000 \text{ N}$ , una altura de la placa  $a = 0.01 \text{ m}$  (10 cm) y una base para la placa  $b = 0.05 \text{ m}$  (5 cm). La distribución de esfuerzos para este modelo fue descrita por Timoshenko et al., 1951 [37] por medio de las ecuaciones en coordenadas polares 23 a la 25:

$$\sigma_r(r, \theta) = \frac{S}{2} \left( 1 - \frac{a^2}{r^2} \right) + \frac{S}{2} \left( 1 + \frac{3a^4}{r^4} - \frac{4a^2}{r^2} \right) \cos 2\theta \quad (23)$$

$$\sigma_\theta(r, \theta) = \frac{S}{2} \left( 1 + \frac{a^2}{r^2} \right) - \frac{S}{2} \left( 1 + \frac{3a^2}{r^4} \right) \cos 2\theta \quad (24)$$

$$\tau_{r\theta}(r, \theta) = -\frac{S}{2} \left( 1 - \frac{3a^4}{r^4} + \frac{2a^2}{r^2} \right) \sin 2\theta \quad (25)$$

Donde:

$$S = \frac{P}{b*H} \quad (26)$$

**Modelo de viga bajo carga distribuida:** Para este modelo, se toma una carga máxima  $q = 3000 \text{ N}$ , un ancho de viga  $c = 0.05 \text{ m}$  (50 mm) y una longitud de la viga  $l = 0.2 \text{ m}$  (20 cm). La distribución de esfuerzos para este modelo fue descrita por Timoshenko et al., 1951 [37] por medio de las ecuaciones en coordenadas rectangulares 27 a la 29:

$$\sigma_x(x, y) = -\frac{q}{2l} (l^2 - x^2)y + \frac{q}{2l} \left( \frac{2}{3}y^3 - \frac{2}{5}c^2y \right) \quad (27)$$

$$\sigma_y(x, y) = -\frac{q}{2I} \left( \frac{1}{3} y^3 - c^2 y + \frac{2}{3} c^3 \right) \quad (28)$$

$$\tau_{xy}(x, y) = -\frac{q}{2I} (c^2 - y^2)x \quad (29)$$

Donde  $I$  corresponde al momento de inercia de la sección transversal de una viga respecto al centroide:

$$I = \frac{2}{3} c^3 \quad (30)$$

**Modelo de viga bajo carga puntual:** Similar al modelo anterior, para este modelo, se toma una carga máxima  $P = 3000 \text{ N}$ , una viga con momento de inercia  $I$  como el expresado en la ecuación (30), un ancho de viga  $c = 0.05 \text{ m}$  ( $50 \text{ mm}$ ) y una longitud de la viga  $l = 0.2 \text{ m}$  ( $20 \text{ cm}$ ). La distribución de esfuerzos para este modelo fue descrita por Timoshenko et al., 1951 [37] por medio de las ecuaciones en coordenadas rectangulares 31 a la 33:

$$\sigma_x = -\frac{Pxy}{I} \quad (31)$$

$$\sigma_y = 0 \quad (32)$$

$$\tau_{xy} = -\frac{P}{I} \frac{1}{2} (c^2 - y^2) \quad (33)$$

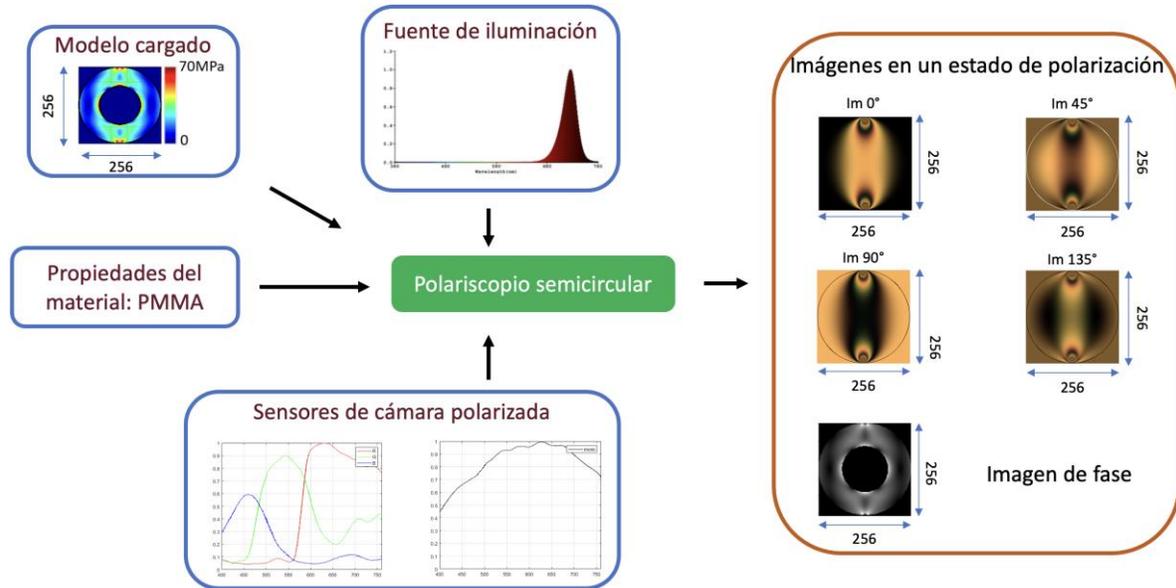
Finalmente, se define que la resolución a utilizar para las imágenes a generar será de  $256 \times 256$  píxeles. Ya teniendo todas estas propiedades, se cuenta con la información suficiente para simular el funcionamiento de un polariscopio y con ello, generar imágenes multipolarizadas de manera analítica. Haría falta únicamente unir todas estas propiedades y ecuaciones en un código que permita realizar el procesamiento y generación de las imágenes. Este código fue en gran parte facilitado por los profesores Juan Carlos Briñez de León, Hermes Fandiño Toro y Alejandro Restrepo Martínez y posteriormente adaptado de acuerdo con las necesidades planteadas en esta tesis. El código utilizado en la generación de estas imágenes puede encontrarse en el Anexo 2.

Por medio de este código, fue posible generar imágenes de tres canales de color (R, G, B) ( $256, 256, 3$ ), para cuatro orientaciones del polarizador:  $10^\circ$ ,  $145^\circ$ ,  $190^\circ$  y  $1135^\circ$ , en dos sensores de cámara polarizada (uno a color y el otro monocromático), para nueve fuentes de iluminación distintas, tipo LED y en 100 pasos de carga distintos, desde los 0N hasta la carga máxima especificada para cada modelo.

Es decir, se tenía un total de 9000 cuartetos de imágenes, donde cada cuarteto correspondía a un mismo modelo, sensor de cámara, fuente de iluminación y carga. Cada imagen dentro del cuarteto correspondía a un estado de polarización distinto.

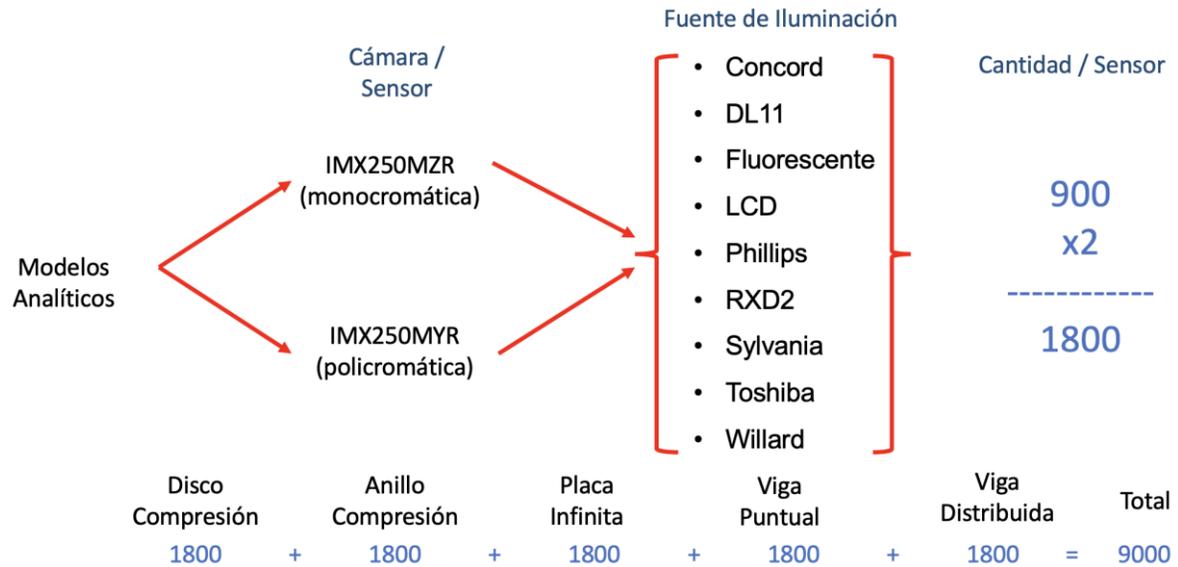
Adicional a esto, con este código también fue posible la construcción de las imágenes objetivo (también llamadas *Ground Truth*), quienes consisten de una única imagen de un único canal (256, 256, 1) representando el campo de esfuerzos distribuido en la geometría cargada y son las imágenes con las que se evaluarán las predicciones de las redes neuronales a entrenar. Es importante mencionar que el campo de esfuerzos generado en una geometría cargada es el mismo, así se varíe el sensor de la cámara y/o la fuente de iluminación.

La imagen 15 resume el proceso aquí descrito:



**Imagen 15.** Proceso analítico de generación de imágenes en cuatro estados de polarización. Adaptado de Briñez et al., 2020 [12].

La imagen 16 explica la distribución de los 9000 cuartetos de imágenes generados, como resultado de la combinatoria de sensores, fuentes de iluminación y modelos analíticos:

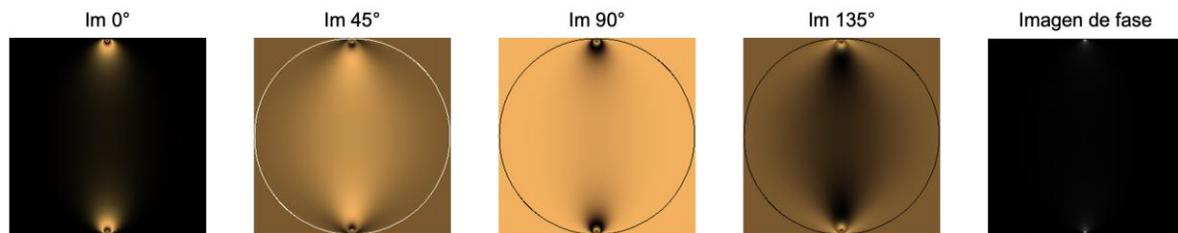


**Imagen 16.** Distribución de generación de imágenes en colección inicial.

A modo de ejemplo, las imágenes 17 a 21 ilustran los cuartetos generados para diferentes configuraciones de geometría, sensor de cámara, fuente de iluminación y carga, en conjunto con su imagen de fase:

**Modelo de disco a compresión:**

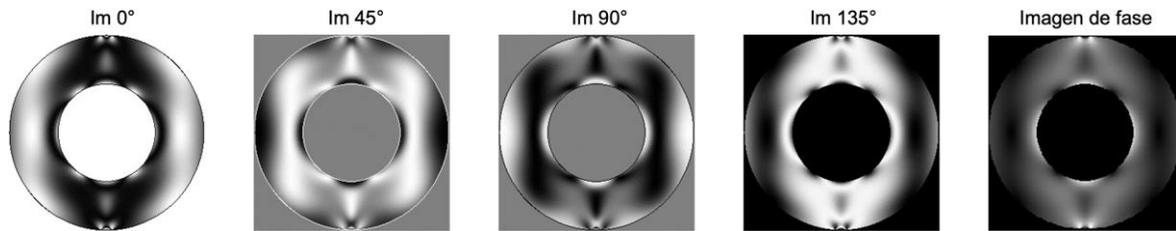
Fuente de iluminación: Cooper DL11 WS WW LED.  
 Sensor: IMX250MYR.  
 Paso de carga número 20, 700 N.



**Imagen 17.** Cuarteto de imágenes por estado de polarización - Geometría de disco a compresión.

**Modelo de anillo a compresión:**

Fuente de iluminación: Fluorescente.  
 Sensor: IMX250MZR.  
 Paso de carga número 40, 1280 N.



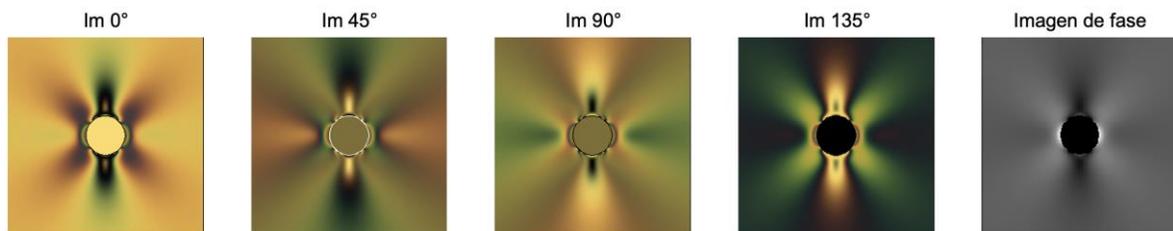
**Imagen 18.** Cuarteto de imágenes por estado de polarización - Geometría de anillo a compresión.

### Modelo de placa infinita:

Fuente de iluminación: Pantalla LCD.

Sensor: IMX250MYR.

Paso de carga número 60, 4200 N.



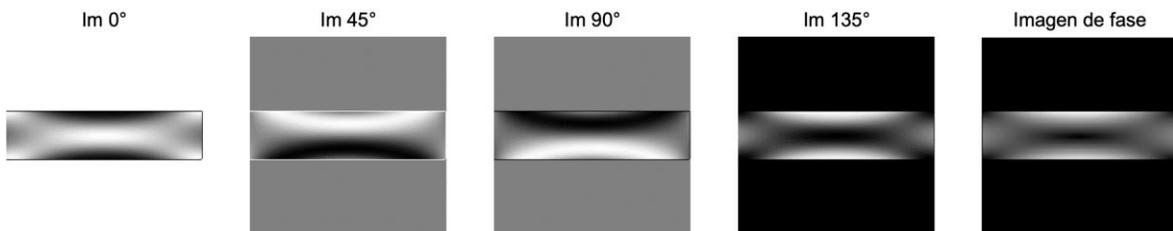
**Imagen 19.** Cuarteto de imágenes por estado de polarización - Geometría de placa infinita.

### Modelo de viga bajo carga distribuida:

Fuente de iluminación: Philips EnduraLED MR16 LED.

Sensor: IMX250MZR.

Paso de carga número 80, 2400 N.



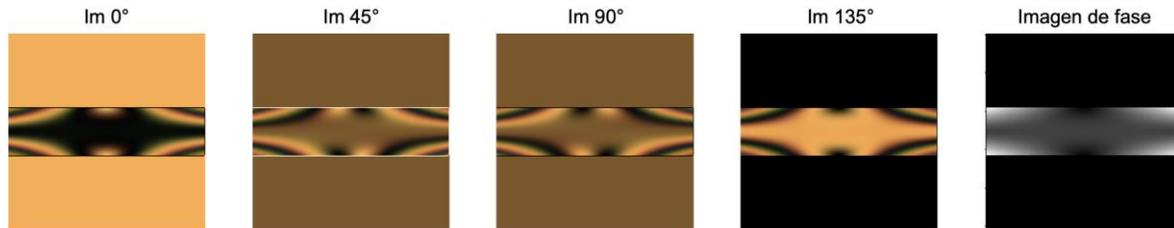
**Imagen 20.** Cuarteto de imágenes por estado de polarización - Geometría de viga carga distribuida.

### Modelo de viga bajo carga puntual:

Fuente de iluminación: Cooper RXD2 RWS WW LED.

Sensor: IMX250MYR.

Paso de carga número 100, 3000 N.



**Imagen 21.** Cuarteto de imágenes por estado de polarización - Geometría de viga carga puntual.

Esta colección inicial de imágenes sirve de insumo para la sección 4.2, en la que se construirán dos representaciones donde se agrupen los diferentes estados de polarización en una única imagen.

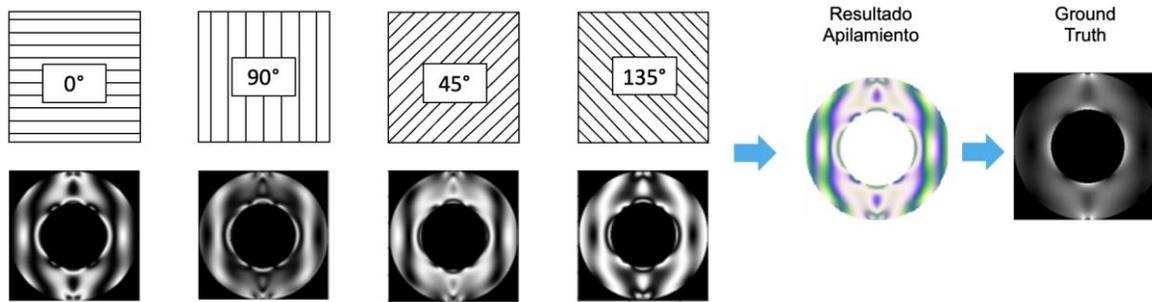
## 4.2 Representación de imágenes multipolarizadas

Con el fin de agrupar las imágenes de fotoelasticidad, de diferentes orientaciones, en una única imagen y construir una colección de imágenes multipolarizadas con el cual se pueda realizar el entrenamiento de las redes neuronales, se proponen dos alternativas que fueron comparadas y evaluadas en el artículo "*Photoelastic and Stokes Images through Deep Convolutional Neural Networks: A Comparison of Stress Fields*": Representación por medio de apilamiento de imágenes (*Stack*) y representación utilizando Vector Stokes.

Ambas representaciones propuestas toman las imágenes con mayor carga (paso de carga número 100), filtran el fondo de las imágenes y realizan un apilamiento del canal G de cada una de las imágenes presentes en este cuarteto, dando como resultado una imagen con carga máxima de dimensiones (256, 256, 4). De esta imagen se saca el valor máximo y con base en este valor se normalizan las demás imágenes que se irán generando.

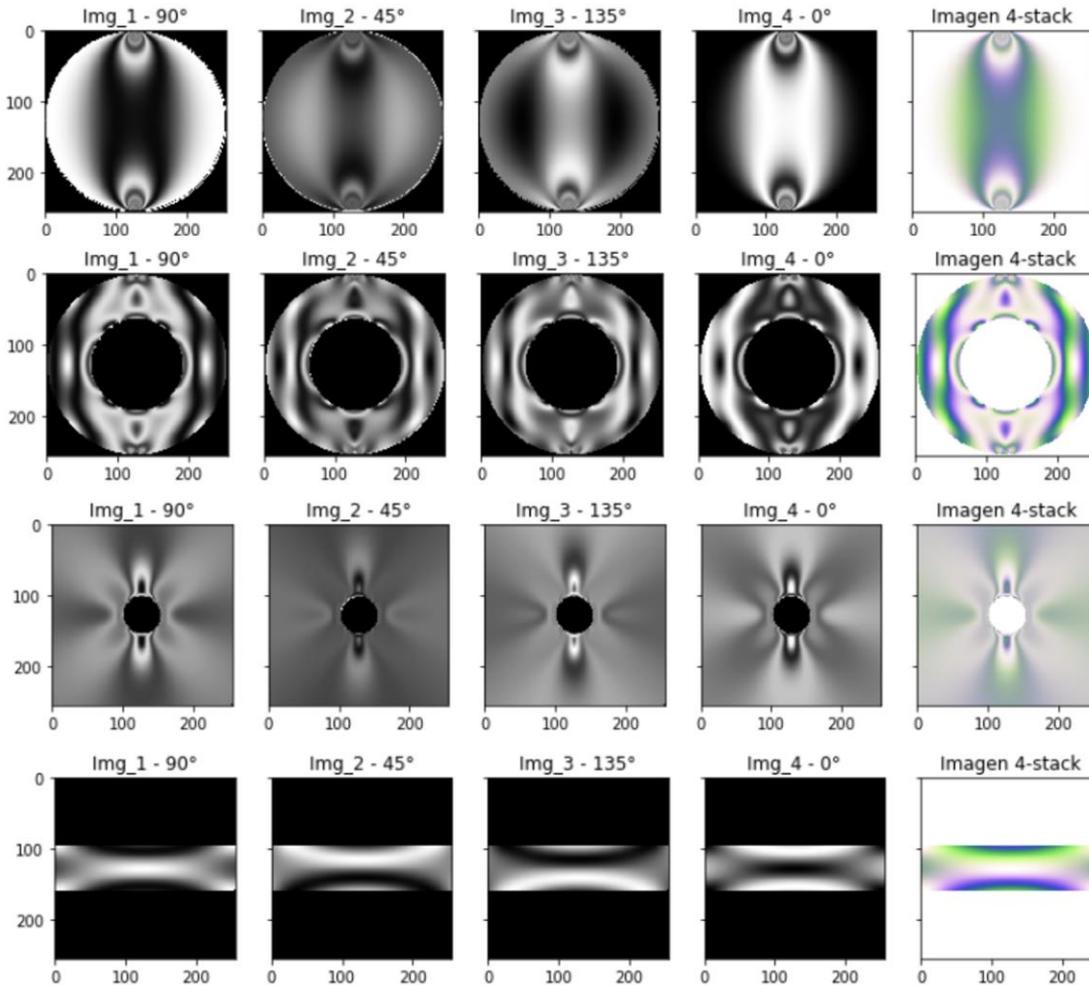
Luego de conocer el valor por el cual se debe normalizar las imágenes, ambas representaciones propuestas toman el canal G de cada una de las imágenes presentes en cada cuarteto  $4 \times (256, 256, 3)$ , quedando así con cuatro imágenes con un solo canal de color  $4 \times (256, 256, 1)$  y procesan estas imágenes para la conformación de una única imagen multipolarizada, que será normalizada al final. La diferencia entre ambas representaciones propuestas radica en la forma en que se combinan estas cuatro últimas imágenes. Así:

**Representación por medio de apilamiento de imágenes (*Stack*):** Esta representación toma las cuatro imágenes del canal verde G y las apila una tras otra, formando una única imagen con cuatro canales de color (256, 256, 4). La imagen 22 resume este proceso:



**Imagen 22.** Generación de imágenes multipolarizadas por apilamiento de imágenes (*Stack*).

De esta manera, cada cuarteto de imágenes en diferentes estados de polarización pasa a conformar una única imagen multipolarizada. Como se contaba con 9000 cuartetos, entonces se forman 9000 imágenes multipolarizadas, 1800 para cada modelo analítico propuesto. La imagen 23 reúne los resultados de apilar cada cuarteto para conformar una imagen multipolarizada, en cada una de las geometrías estudiadas:



**Imagen 23.** Representación por apilamiento de imágenes (*Stack*) en diferentes geometrías.

**Representación por medio de Vector Stokes:** El enfoque más común para describir el estado de polarización de la luz emitida o reflejada es medir los parámetros de Stokes:  $S_0$ ,  $S_1$  y  $S_2$ .

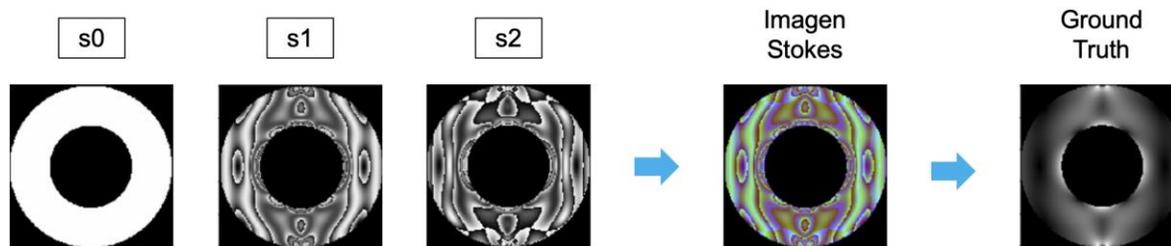
Para aplicaciones de imagen, estos parámetros, se determinan píxel a píxel para reconstruir una imagen 2D. Los parámetros de Stokes se determinan midiendo la intensidad de la radiancia, que es emitida por un polarizador que se orienta en varios ángulos para medir la magnitud de un estado dado (Gurton et al., 2014) [34].

En la literatura se pueden encontrar varias ecuaciones para construir las imágenes de Stokes a través de  $S_0$ ,  $S_1$  y  $S_2$ . En este estudio, se decide seguir las ecuaciones establecidas por Vedel et al., 2010 [63], tal como se enuncia en la ecuación 34:

$$S = \begin{bmatrix} S_0 \\ S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} * \begin{bmatrix} I_{0^\circ} \\ I_{90^\circ} \\ I_{45^\circ} \\ I_{135^\circ} \end{bmatrix} \quad (34)$$

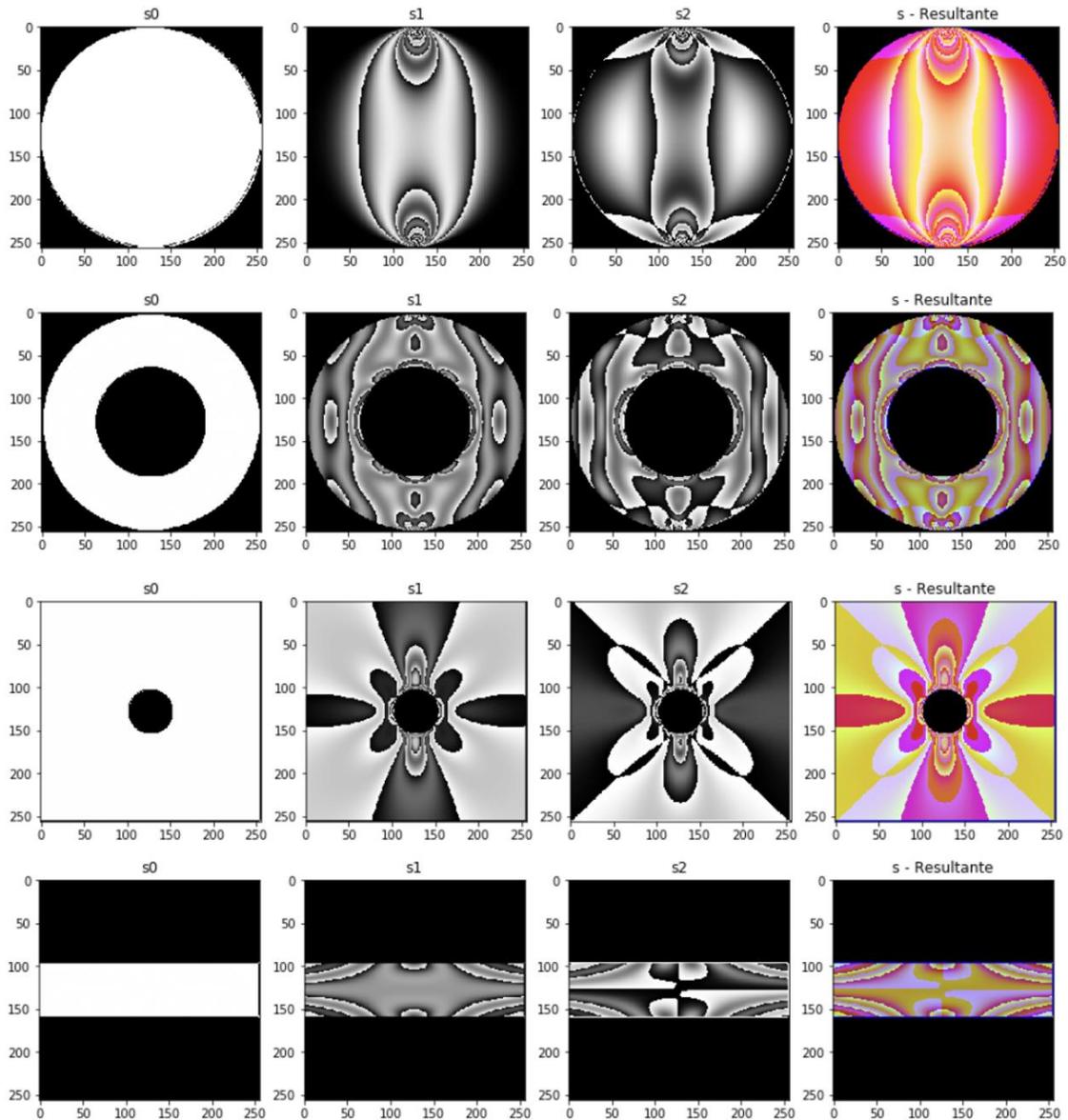
Donde  $I_{0^\circ}$ ,  $I_{45^\circ}$ ,  $I_{90^\circ}$  e  $I_{135^\circ}$  corresponden al canal G de cada de las imágenes en los diferentes estados de polarización.

Una vez se forman  $S_0$ ,  $S_1$  y  $S_2$ , cada parámetro de Stokes tiene una dimensión de (256, 256, 1). Con estos parámetros se realiza un apilamiento dando como resultado una única imagen de 3 canales (256, 256, 3), donde cada canal corresponde a un parámetro de Stokes. La imagen 24 ilustra este proceso:



**Imagen 24.** Generación de imágenes multipolarizadas por Vector Stokes.

De manera similar, cada cuarteto de imágenes en diferentes estados de polarización pasa a conformar una única imagen multipolarizada por vector Stokes. Como se contaba con 9000 cuartetos, entonces se forman 9000 imágenes multipolarizadas, 1800 para cada modelo analítico propuesto. La imagen 25 reúne los resultados de formar vectores Stokes cada cuarteto para conformar una imagen multipolarizada, en cada una de las geometrías estudiadas:



**Imagen 25.** Representación por vector Stokes en diferentes geometrías.

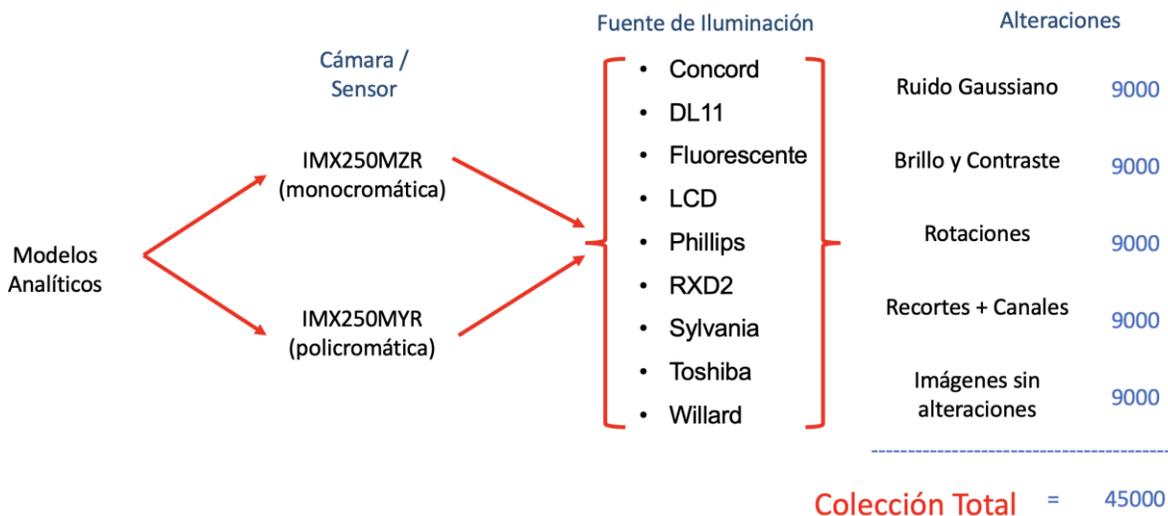
Aquí también se hace importante aclarar que, el hecho de utilizar diferentes representaciones de las imágenes con el ánimo de conformar imágenes multipolarizadas, no modifica para nada la imagen a predecir por los modelos de red neuronal, es decir, la distribución de campo de esfuerzos. Simplemente son distintas maneras de organizar y representar la información contenida en las imágenes.

### 4.3 Aumento de datos

Con la finalidad de que el entrenamiento de las redes neuronales fuera más versátil y capaz de predecir ante posibles variaciones en las imágenes, se hizo necesario realizar un proceso de aumento de datos que consiste en tomar las 9000 imágenes sin alteraciones,

de ambas colecciones (Stack y Stokes) y aplicarles algunas modificaciones clásicas, utilizadas en el procesamiento de imágenes, tales como: adición de ruido gaussiano, modificación del brillo y el contraste de la imagen, rotaciones, seccionamiento de imágenes (también llamado *cropping*) o composición de imágenes a partir de un canal diferente al canal G, que se utilizó para conformar las imágenes iniciales.

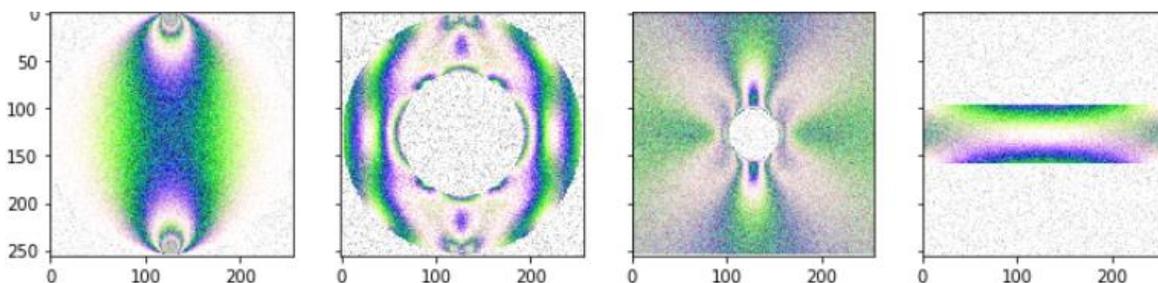
El proceso consistió entonces en que para cada una de las 9000 imágenes de cada colección se aplicaron cada una de estas modificaciones, dando como resultado 9000 imágenes adicionales para cada alteración. De esta manera, cada colección de imágenes tiene un total de 45000 imágenes, incluyendo las imágenes iniciales sin alteraciones. La imagen 26 resume el proceso de aumento de datos:



**Imagen 26.** Distribución total de imágenes por colección, con proceso de aumento de datos.

A continuación, se detallan una a una las alteraciones realizadas a las imágenes en el proceso de aumento de datos:

**Ruido Gaussiano:** Se generó una función que tomaba las imágenes sin alteraciones de ambas colecciones y les aplicaba ruido siguiendo una distribución gaussiana. Las imágenes 27 y 28 ilustran el resultado para cada una de las colecciones:



**Imagen 27.** Alteración por ruido gaussiano en la colección de imágenes apiladas (*Stack*).

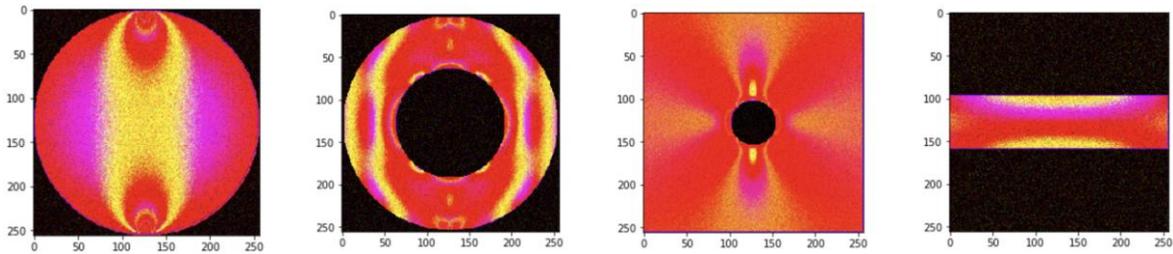


Imagen 28. Alteración por ruido gaussiano en la colección de imágenes Stokes.

**Brillo y contraste:** La librería openCV cuenta con una funcionalidad llamada *convertScaleAbs*, que se utiliza para convertir una imagen de un tipo de datos a otro y luego realizar una operación de valor absoluto en cada píxel de la imagen resultante.

El uso de *convertScaleAbs* es común cuando se desea realizar operaciones aritméticas en imágenes, como sumar o restar imágenes de diferentes tipos de datos, ajustar el brillo y el contraste, o aplicar operaciones matemáticas complejas. Tiene como parámetros a *alpha* que es la escala y *beta* que es el desplazamiento.

En este caso, se utilizó esta librería con valores aleatorios para *alpha* y *beta*, con límites entre 1.0 y 3.0 para *alpha* y entre 0 y 100 para *beta*. *Alpha* controla las variaciones en el contraste de una imagen mientras que *beta* controla las variaciones en el brillo. Las imágenes 29 y 30 ilustran el resultado para cada una de las colecciones:

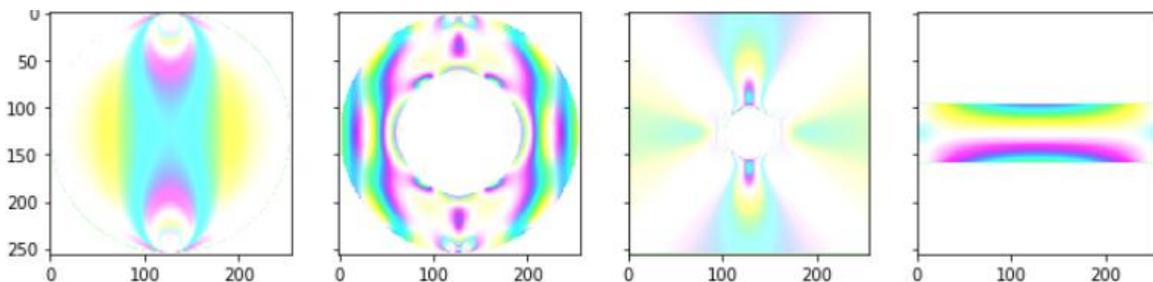


Imagen 29. Alteración por brillo y contraste en la colección de imágenes apiladas (*Stack*).

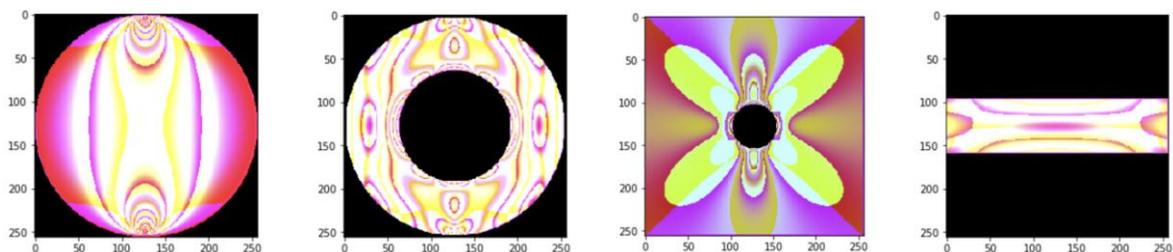


Imagen 30. Alteración por brillo y contraste en la colección de imágenes Stokes.

**Rotaciones:** Nuevamente se recurre a la librería openCV pero en este caso a sus funcionalidades *getRotationMatrix2D* y *warpAffine*. Por medio de ellas, se realiza una rotación desde los  $10^\circ$  hasta los  $170^\circ$  de cada uno de los modelos, como se ilustra en las imágenes 31 y 32:

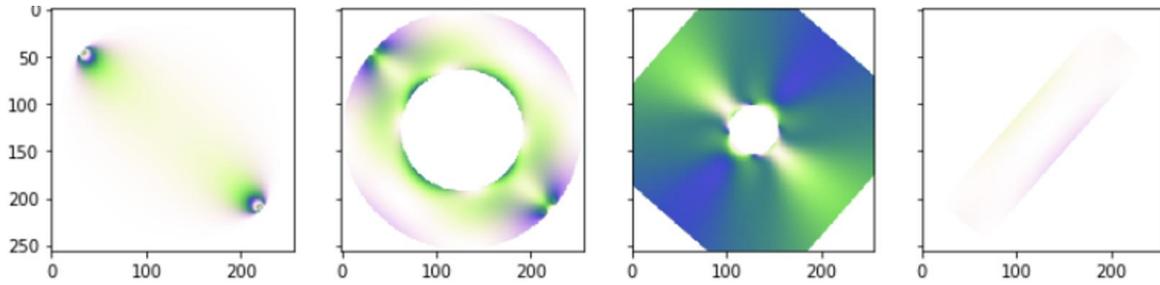


Imagen 31. Alteración por rotaciones en la colección de imágenes apiladas (*Stack*).

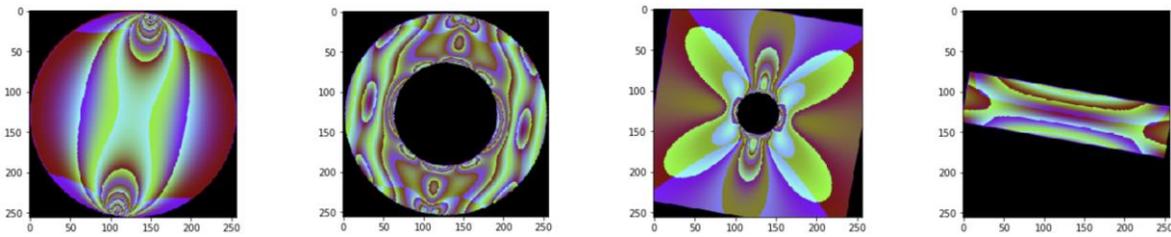


Imagen 32. Alteración por rotaciones en la colección de imágenes Stokes.

**Recortes y canales:** De manera aleatoria, se selecciona uno de los dos canales faltantes por utilizar: canal rojo R o canal azul B y, posterior a esto, se realiza un seccionamiento de la imagen, dividiéndola en cuatro cuadrantes y tomando solo uno de ellos como la imagen total. Las imágenes 33 y 34 ilustran este proceso:

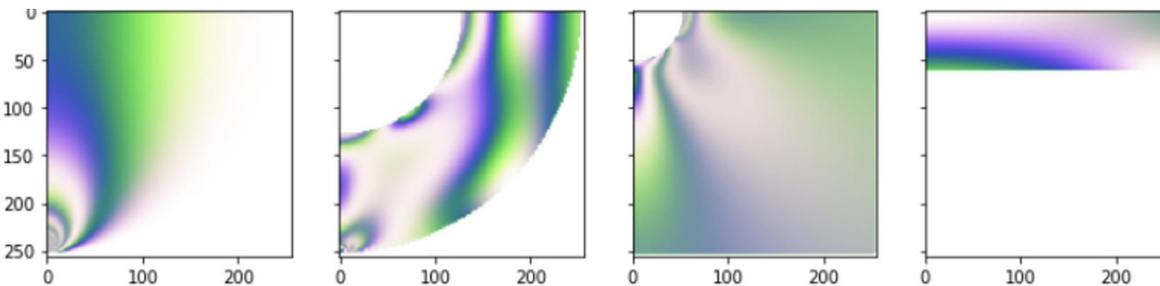


Imagen 33. Alteración por recortes y canales en la colección de imágenes apiladas (*Stack*).

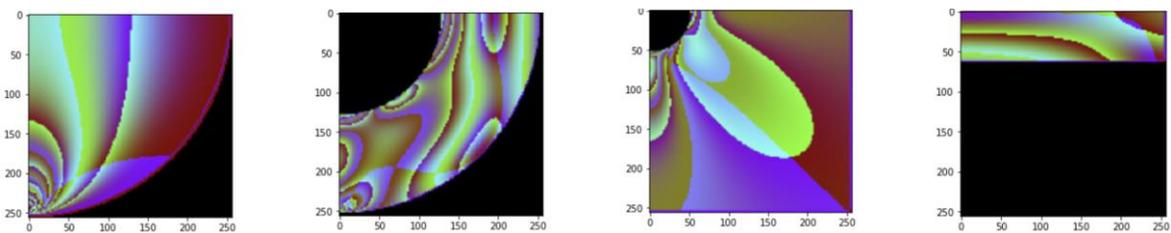


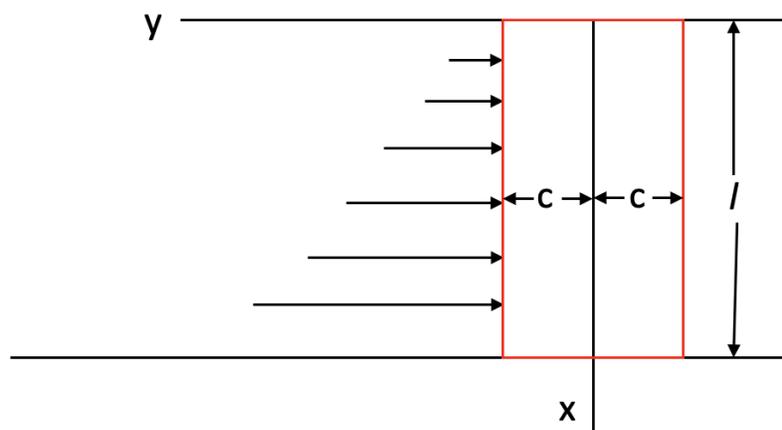
Imagen 34. Alteración por recortes y canales en la colección de imágenes Stokes.

## 4.4 Modelos adicionales para validación

Una vez se tienen las colecciones de entrenamiento de las redes neuronales, se hace necesario contar con algunos modelos adicionales que sirvan para validar y evaluar el comportamiento de los modelos a entrenar ante imágenes nunca antes vistas, que de una u otra manera repliquen la evaluación que se haría con imágenes reales tomadas con cámaras multipolarizadas.

Para lograr este objetivo, se recurre nuevamente a algunos modelos clásicos estudiados por diferentes autores, tales como:

**Modelo de flexión de viga sometida a carga variable continua:** como podría ser el caso en el que se desean determinar los esfuerzos que origina una presión hidrostática en una estructura vertical empotrada. La imagen 35 ilustra este modelo:



**Imagen 35.** Diagrama de modelo de flexión de viga sometida a carga variable continua. Adaptado de Timoshenko et al., 1951 [37].

Las ecuaciones que rigen este modelo se presentan a continuación, en coordenadas cartesianas. (Ecuaciones 35 a la 37) (Timoshenko et al., 1951 [37]):

$$\sigma_x = \frac{qx^3y}{4c^3} + \frac{q}{4c^3} \left( -2xy^3 + \frac{6}{5}c^2xy \right) \quad (35)$$

$$\sigma_y = -\frac{qx}{2} + qx \left( \frac{y^3}{4c^3} - \frac{3y}{4c} \right) \quad (36)$$

$$\tau_{xy} = \frac{3qx^2}{8c^3} (c^2 - y^2) - \frac{q}{8c^3} (c^4 - y^4) + \frac{3q}{20c} (c^2 - y^2) \quad (37)$$

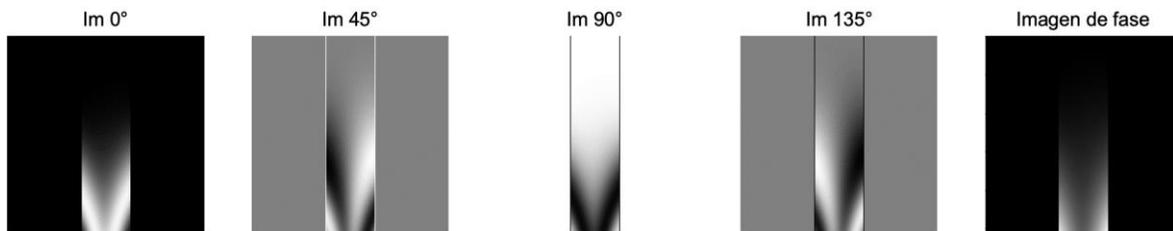
Donde  $q$  representa el peso de la unidad de volumen del fluido, de manera que la intensidad de la carga a una profundidad  $x$  es  $qx$ . Por su parte,  $c$  corresponde a la distancia al eje neutro de la viga.

Con estas ecuaciones se realiza un proceso similar al explicado en la sección 4.1 y se obtienen las imágenes en cada uno de los estados de polarización, al aplicar una carga máxima de 7000 N. La imagen 36 ilustra el proceso realizado.

Fuente de iluminación: Sylvania UltraLED Par 30LN.

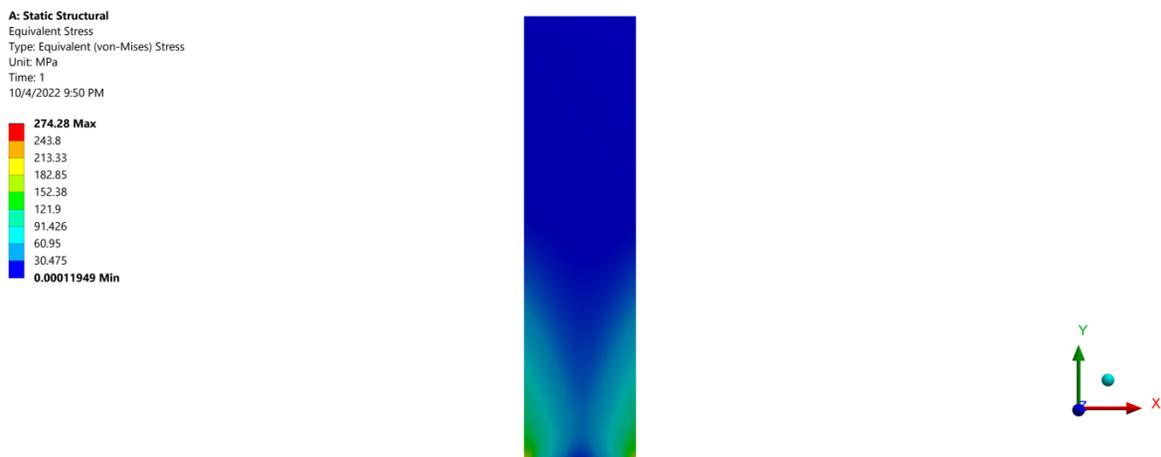
Sensor: IMX250MZR.

Paso de carga número 25, 1750 N.



**Imagen 36.** Cuarteto de imágenes por estado de polarización - Geometría de estructura vertical bajo presión continua.

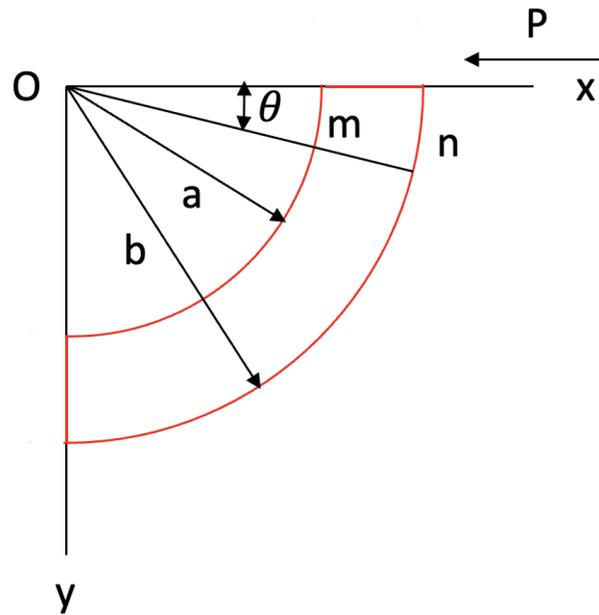
A manera de validación, se realizó una simulación mediante el software de elementos finitos Ansys Workbench 2019. En él, se recreó la geometría estudiada y se aplicó una carga similar a la aplicada en este modelo. Se graficó el resultado de la diferencia de esfuerzos principales sobre la geometría. La imagen 37 ilustra este resultado obtenido:



**Imagen 37.** Simulación de estructura vertical bajo presión lateral en Ansys Workbench.

Puede observarse que la simulación arroja un perfil de esfuerzos muy similar al que se genera en las imágenes polarizadas.

**Modelo de ménsula curva cargada en su extremo libre:** Una pieza de sección rectangular estrecha y directriz circular, está empotrada en su extremo inferior, siendo sometida en el extremo superior a la acción de una fuerza radial  $P$ . La imagen 38 ilustra este modelo:



**Imagen 38.** Diagrama de modelo de ménsula cargada en extremo libre. Adaptado de Timoshenko et al., 1951 [37].

Las ecuaciones que rigen este modelo se presentan a continuación, en coordenadas polares. (Ecuaciones 38 a la 40) (Timoshenko et al., 1951 [37]):

$$\sigma_r = \frac{1}{r} \frac{\partial \phi}{\partial r} + \frac{1}{r^2} \frac{\partial^2 \phi}{\partial \theta^2} = \left( 2Ar - \frac{2B}{r^3} + \frac{D}{r} \right) \sin \theta \quad (38)$$

$$\sigma_\theta = \frac{\partial^2 \phi}{\partial r^2} = \left( 6Ar + \frac{2B}{r^3} + \frac{D}{r} \right) \sin \theta \quad (39)$$

$$\tau_{r\theta} = -\frac{\partial}{\partial r} \left( \frac{1}{r} \frac{\partial \phi}{\partial \theta} \right) = -\left( 2Ar - \frac{2B}{r^3} + \frac{D}{r} \right) \cos \theta \quad (40)$$

Con:

$$A = \frac{P}{2N} \quad (41)$$

$$B = -\frac{Pa^2b^2}{2N} \quad (42)$$

$$D = -\frac{P(a^2+b^2)}{N} \quad (43)$$

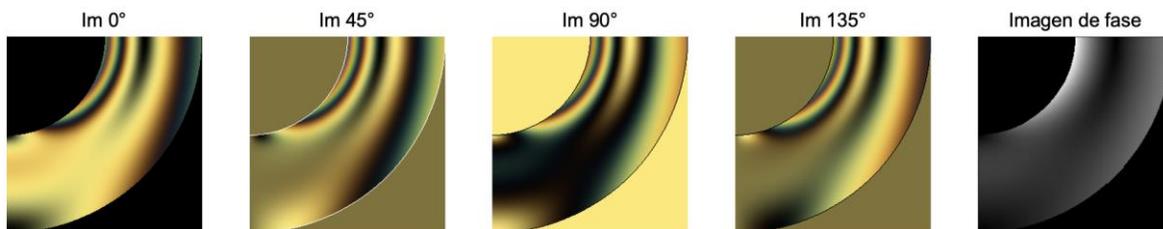
$$N = a^2 - b^2 + (a^2 + b^2) \log \frac{b}{a} \quad (44)$$

Donde  $a$  representa el diámetro interior de la ménsula y  $b$  el diámetro exterior. De manera similar, se generan las imágenes en cada uno de los estados de polarización, al aplicar una carga máxima de 10000 N. La imagen 39 ilustra el proceso realizado.

Fuente de iluminación: Toshiba ECore Par30s 23Deg LED.

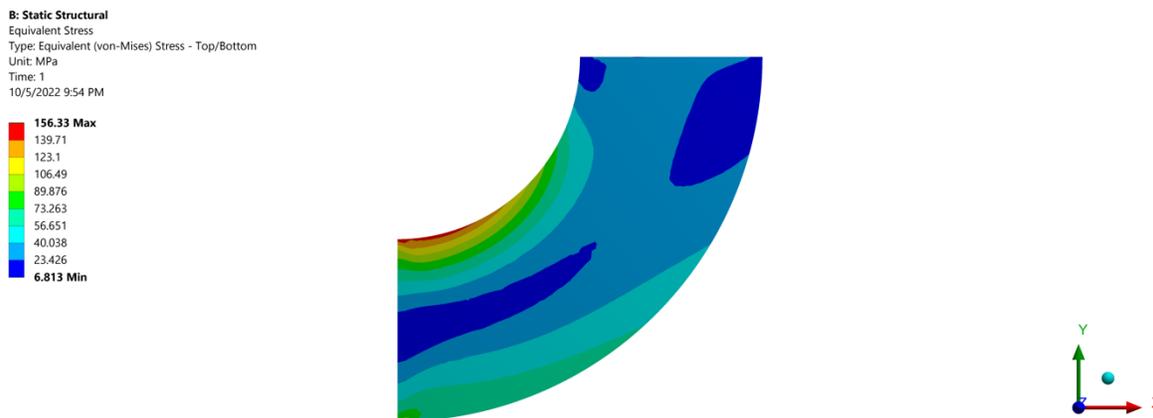
Sensor: IMX250MYR.

Paso de carga número 75, 7500 N.



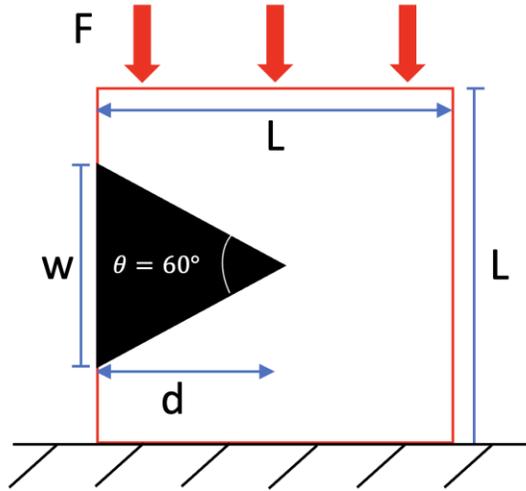
**Imagen 39.** Cuarteto de imágenes por estado de polarización - Geometría de ménsula curva cargada en su extremo libre.

De manera similar, se simuló esta geometría y sus cargas con ayuda del software Ansys Workbench. Los resultados se muestran en la imagen 40:



**Imagen 40.** Simulación de ménsula curva cargada en su extremo libre en Ansys Workbench.

**Modelo de placa rectangular finita con muesca tipo 'V' bajo compresión uniaxial:** Finalmente, para este modelo, Kramer et al., 2009 [38] presenta las ecuaciones 45 a 47 en coordenadas polares, para describir la distribución de esfuerzos en el modelo representado por la imagen 41:



**Imagen 41.** Diagrama de modelo de placa rectangular finita con muesca tipo 'V' bajo compresión uniaxial. Adaptado de Kramer et al., 2009 [38].

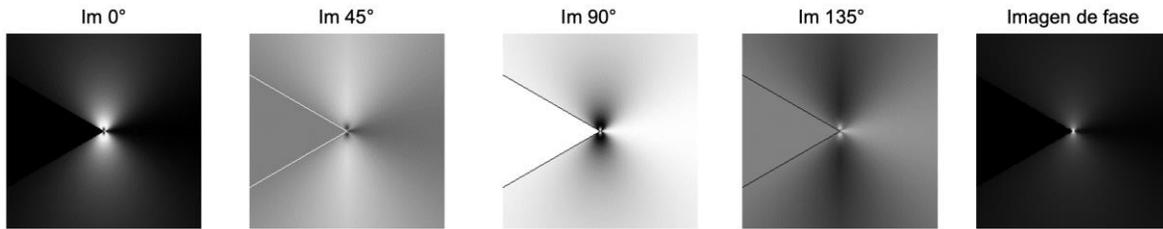
$$\sigma_{rr}(r, \theta) = \frac{C_f * \sigma_{app} * d^{1-\lambda_0}}{(r^{1-\lambda_0})} \left\{ \begin{array}{l} -\lambda_0(\lambda_0 + 1) * \cos((\lambda_0 + 1) * \theta) + \lambda_0(\lambda_0 - 3) * \\ \frac{\cos((\lambda_0+1)*\frac{\xi}{2})}{\cos((\lambda_0-1)*\frac{\xi}{2})} * \cos((\lambda_0 - 1) * \theta) \end{array} \right\} \quad (45)$$

$$\sigma_{\theta\theta}(r, \theta) = \frac{C_f * \sigma_{app} * d^{1-\lambda_0}(\lambda_0+1)}{(r^{1-\lambda_0})} \left\{ \begin{array}{l} \cos((\lambda_0 + 1) * \theta) - \\ \frac{\cos((\lambda_0+1)*\frac{\xi}{2})}{\cos((\lambda_0-1)*\frac{\xi}{2})} * \cos((\lambda_0 - 1) * \theta) \end{array} \right\} \quad (46)$$

$$\sigma_{r\theta}(r, \theta) = \frac{C_f * \sigma_{app} * d^{1-\lambda_0} \lambda_0}{(r^{1-\lambda_0})} \left\{ \begin{array}{l} (\lambda_0 + 1) * \sin((\lambda_0 + 1) * \theta) - \\ (\lambda_0 - 1) * \frac{\cos((\lambda_0+1)*\frac{\xi}{2})}{\cos((\lambda_0-1)*\frac{\xi}{2})} * \sin((\lambda_0 - 1) * \theta) \end{array} \right\} \quad (47)$$

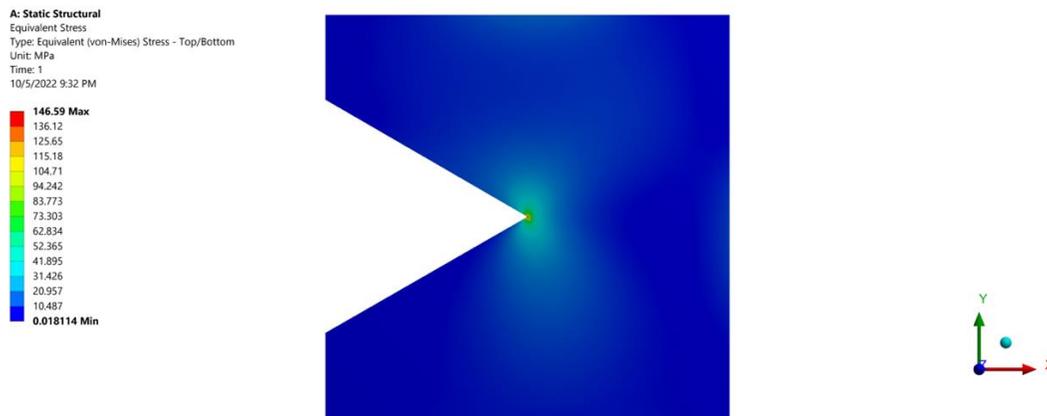
Donde  $\sigma_{app} = -1.84$  MPa es el esfuerzo aplicado en el campo lejano,  $\lambda_0 = 0.512221$  para una muesca en 'V' de  $60^\circ$ ,  $C_f = 0.92$  es un parámetro de ajuste que depende de la geometría de la muesca,  $\xi = 5\pi/3$  es el material que queda después de recortar la muesca en 'V' y  $d$  es la profundidad de la muesca 'V'.

Al utilizar estas ecuaciones para construir las imágenes en cada uno de los estados de polarización y aplicar una carga máxima de 7000 N se obtiene el resultado mostrado por la imagen 42:



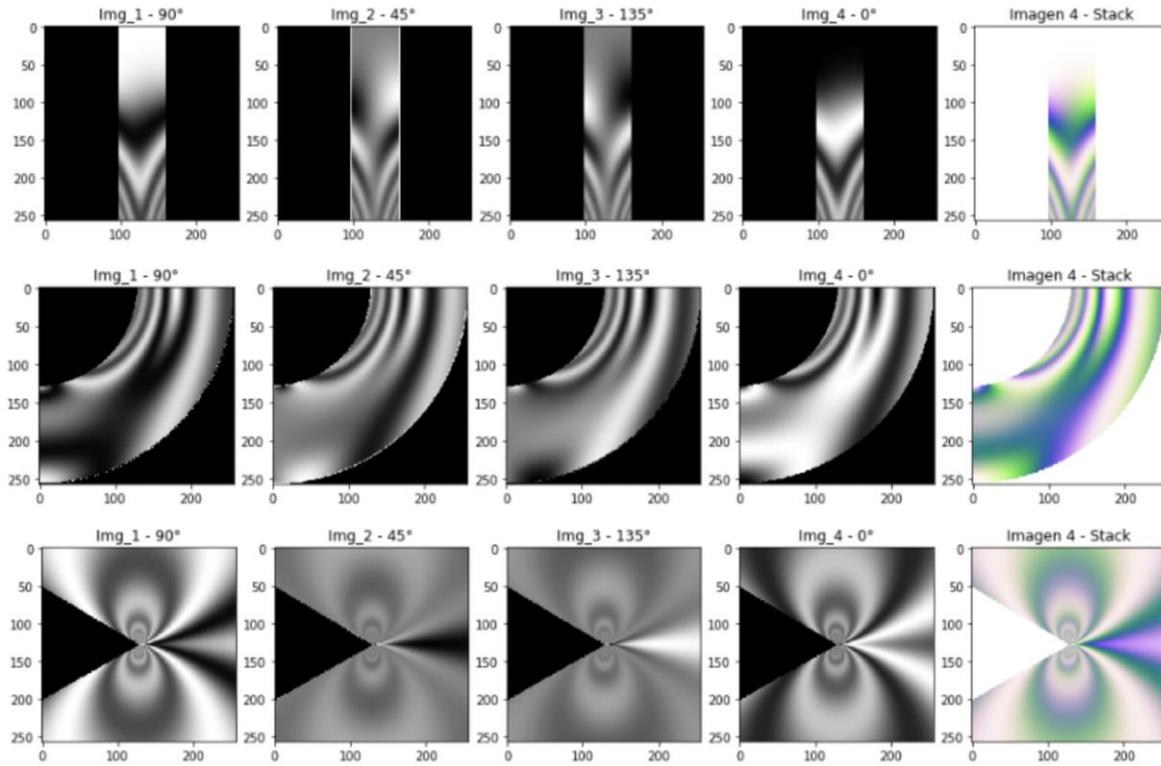
**Imagen 42.** Cuarteto de imágenes por estado de polarización - Geometría de placa rectangular finita con muesca tipo 'V' bajo compresión uniaxial.

De manera similar, la figura 43 ilustra el perfil obtenido al realizar la simulación con Ansys:

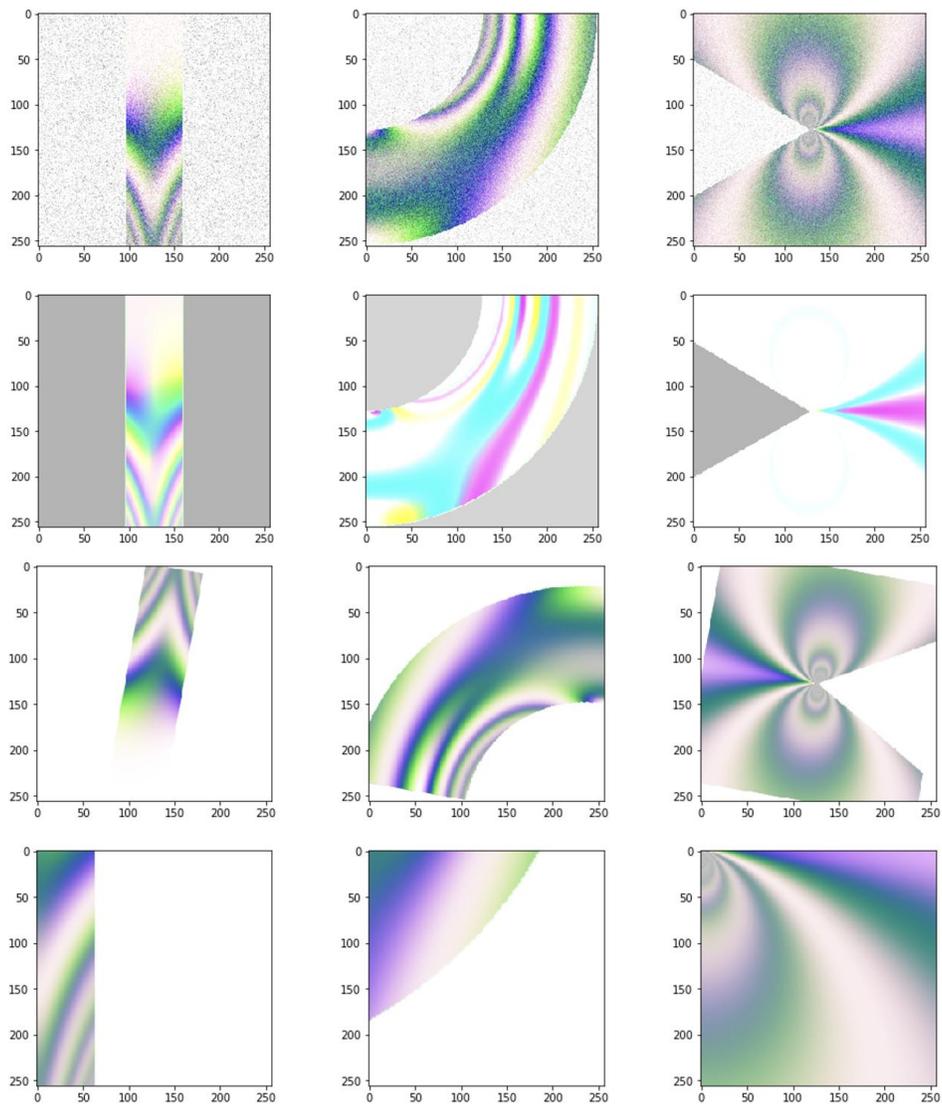


**Imagen 43.** Simulación de placa rectangular finita con muesca tipo 'V' bajo compresión uniaxial en Ansys Workbench.

Finalmente, siguiendo los pasos realizados con los modelos iniciales, se construyen las imágenes multipolarizadas utilizando la representación apilada y con ello, también se realizan las alteraciones para aumento de datos. Las imágenes 44 y 45 ilustran estos resultados:



**Imagen 44.** Representación por apilamiento de imágenes (*Stack*) en geometrías de validación.



**Imagen 45.** Alteraciones para aumento de datos en la colección de imágenes de validación por representación apilada (*Stack*).

## 5. Desarrollo y Evaluación de Arquitecturas de Redes Neuronales

En este capítulo, se describe el proceso de selección, entrenamiento, evaluación y optimización de una arquitectura de red neuronal, lo cual atiende los objetivos específicos II y III de esta tesis.

Las métricas de calidad de imagen (objetivo III) son frecuentemente utilizadas dentro de la etapa de desarrollo de las redes, con el fin de evaluar su desempeño y así, poder llevar a cabo una selección, refinamiento y optimización de las redes estudiadas.

El capítulo se estructura de la siguiente manera: Inicialmente, se realiza una exploración sobre algunas de las arquitecturas clásicas de red neuronal que han sido utilizadas en la extracción de campos de esfuerzo. Luego de esto, se estudia la inclusión de métricas de calidad de imagen para realizar la evaluación de los modelos propuestos. Posteriormente, se replican algunas de las arquitecturas propuestas y se entrenan unas primeras implementaciones utilizando la colección de imágenes apiladas (*Stack*) para lograr seleccionar una arquitectura base, partiendo de los resultados obtenidos en las métricas.

Una vez se ha elegido la arquitectura base, se comienza con un refinamiento de la misma, al optimizar los resultados obtenidos mediante experimentos con varias funciones de pérdida de otros autores y funciones de pérdida propias.

A continuación, se combinan métodos iterativos manuales en conjunto con la librería *Keras optimizer* para optimizar componentes e hiper-parámetros de la arquitectura y desarrollar así la *MultipolarNet*, una red neuronal optimizada para el problema estudiado en esta tesis.

### 5.1 Arquitecturas consideradas

Como punto de partida para el desarrollo de este objetivo, se toman algunas de las arquitecturas referenciadas en el estado del arte, se analiza su estructura y características:

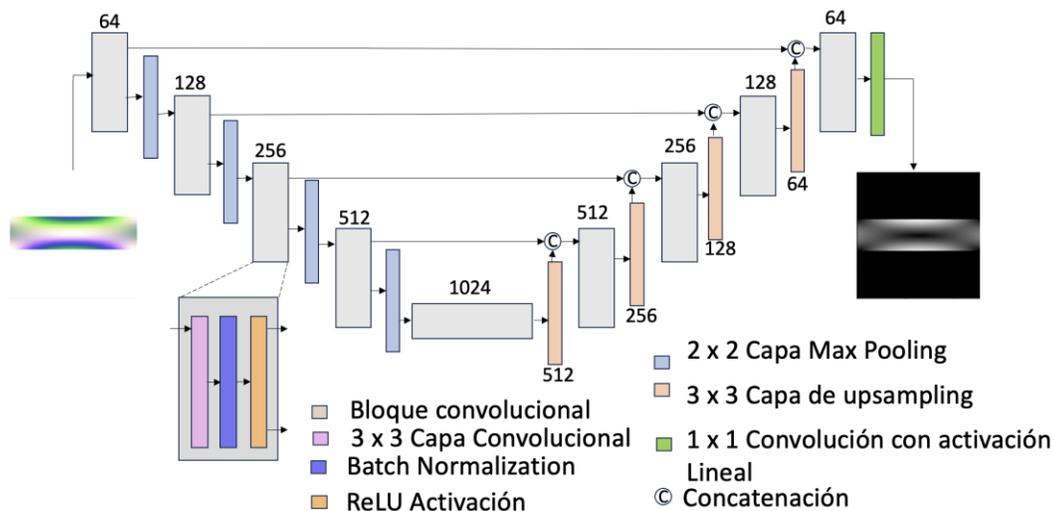
#### 5.1.1 UNet

Propuesta por Ronnenberg et al., 2015 [41], esta arquitectura es ampliamente utilizada en problemas de segmentación, regresión y traducción de imágenes debido a su capacidad para captar características relevantes a diferentes escalas. Su arquitectura consiste de dos bloques anidados de capas de redes neuronales: un bloque codificador (*encoder*) y uno decodificador (*decoder*).

El bloque codificador está conformado por una serie de capas convolucionales 2D (*Conv2D*) que normalmente van seguidas de capas *BatchNormalization* y submuestreo (*MaxPooling2D*) que capturan progresivamente características de nivel superior, lo cual reduce la resolución espacial de la imagen, pero aumenta la capacidad de extraer características. Este bloque normalmente es activado por una función de activación (*Activation*) tipo *ReLU*, más no es excluyente.

Por su lado, el bloque decodificador está conformado por capas de convolución (*Conv2D*) y deconvolución (*UpSampling2D*) para aumentar la resolución espacial y reconstruir una máscara de segmentación precisa o una imagen de regresión. De igual manera, este bloque normalmente es activado también por una función de activación (*Activation*) tipo *ReLU*.

Se presentan conexiones de salto entre los bloques codificador y decodificador que permiten pasar información de bajo nivel directamente al decodificador, lo que ayuda a fusionar características a diferentes escalas y a mejorar la precisión de la segmentación/regresión. La imagen 46 representa una arquitectura UNet adaptada al problema de extracción de campos de esfuerzo en imágenes multipolarizadas:



**Imagen 46.** Esquema UNet clásica adaptada al problema de extracción de esfuerzos en imágenes multipolarizadas. Adaptado de Perera et al., 2020 [14].

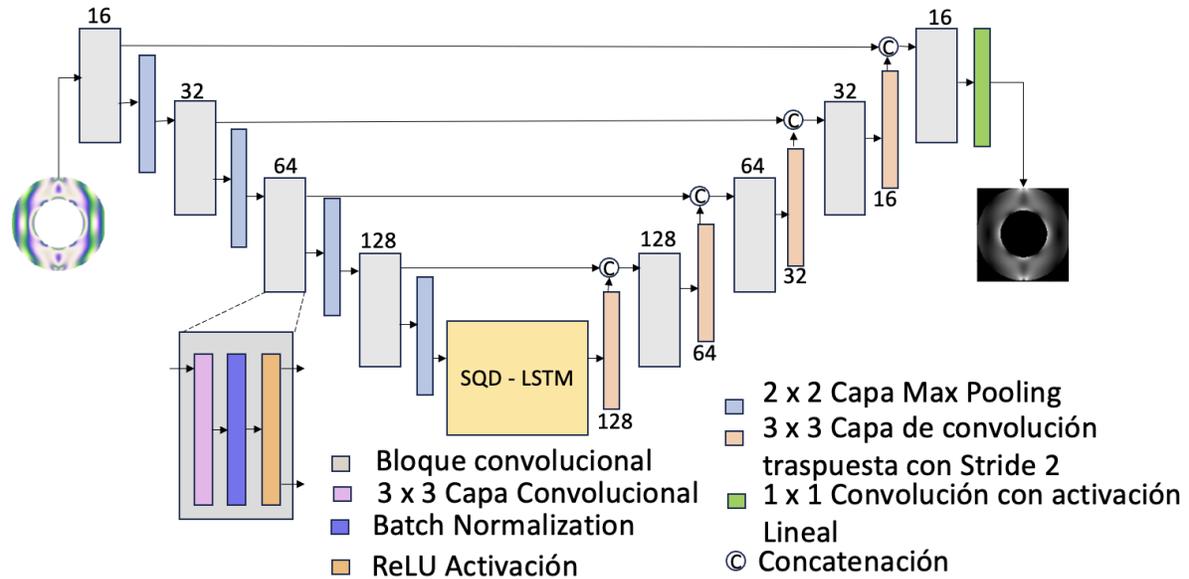
Al ser una arquitectura clásica, la UNet ha sido adaptada para diferentes casos de uso, modificando sus capas, cantidad de neuronas, funciones de pérdida, entre otros hiperparámetros.

### 5.1.2 UNet con capas LSTM

Algunos años más tarde, Perera et al., 2020 [14] retoman la arquitectura UNet clásica y añaden capas intermedias con memoria a corto plazo cuatridireccional espacial SQD-LSTM, argumentando que con la adición de estas capas, se puede sortear la dificultad inherente a las CNN de aprender dependencias espaciales globales, que son vitales para recuperar la fase real.

Además de esto, este autor resulta de gran importancia, al aportar no solo una nueva arquitectura sino también una función de pérdida compuesta específica del problema para entrenar esta red, la cual será evaluada más adelante en este mismo capítulo.

La red propuesta obtiene mejores resultados que los métodos existentes en condiciones de ruido intenso y emplea un tiempo de cálculo significativamente menor. La imagen 47 ilustra esta arquitectura:



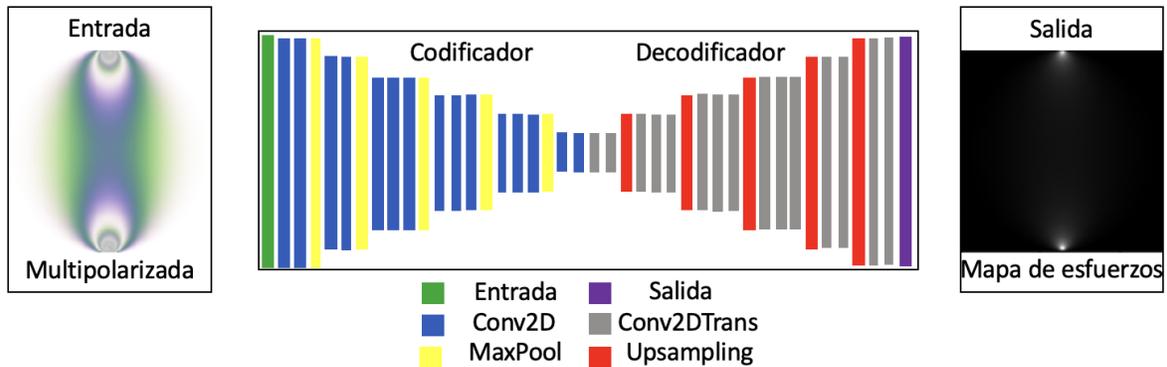
**Imagen 47.** Esquema UNet con capas LSTM adaptada al problema de extracción de esfuerzos en imágenes multipolarizadas. Adaptado de Perera et al., 2020 [14].

### 5.1.3 StressNet

Durante el mismo año, Briñez et al., 2020 [3] desarrollaron la denominada *StressNet*, una red neuronal robusta enfocada en la demodulación de patrones de franjas en una superficie continua de esfuerzos con un buen rendimiento, pero donde queda latente una profundización en las diferencias en el esfuerzo máximo.

Esta arquitectura es de las más robustas, al asemejarse a la arquitectura básica de la UNet, pero con un aumento significativo en el número de capas convolucionales. Su estructura codificadora se compone de capas convolucionales 2D (*Conv2D*) anidadas con capas de normalización *BatchNormalization*, funciones de activación (*Activation*) tipo ReLU y capas de submuestreo (*MaxPooling2D*).

Por su parte, la etapa decodificadora se compone de capas convolucionales traspuestas (*Conv2DTranspose*) anidadas con capas de normalización *BatchNormalization*, funciones de activación (*Activation*) tipo ReLU y capas de muestreo ascendente (*UpSampling2D*). El error cuadrático medio (*MSE*) es utilizado como función de pérdida. La imagen 48 ilustra esta arquitectura:

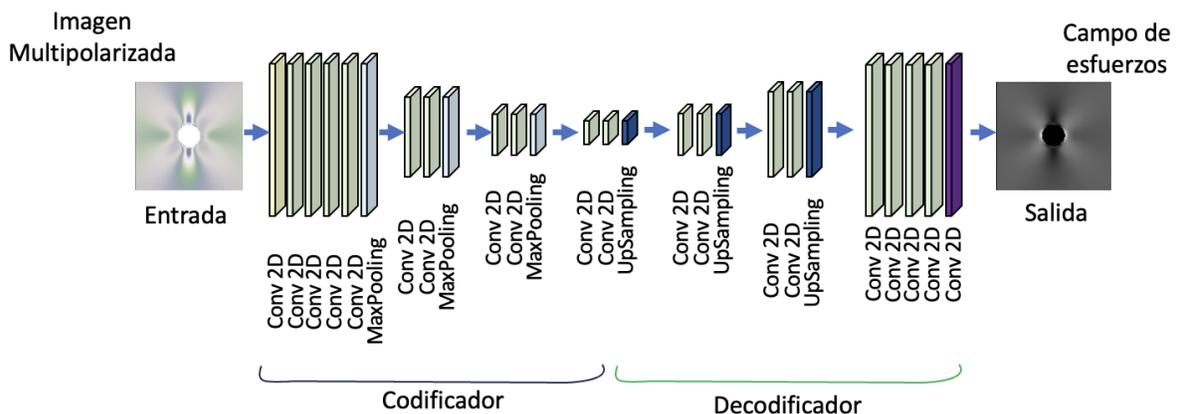


**Imagen 48.** Esquema StressNet y componentes. Adaptado de Briñez et al., 2020 [3].

### 5.1.4 PhotoelastNet

Dos años después, Briñez et al., 2022 [2] desarrollan la denominada *PhotoelastNet*, especializada en la reconstrucción de fase, sin la necesidad de complicados montajes ópticos. Evita problemas relacionados con inconsistencias, ambigüedades o franjas isoclínicas. Esta red es mucho más ligera que la *StressNet* y es capaz de alcanzar métricas de calidad de imagen similares.

Su estructura general es básicamente semejante a la de la *StressNet*, pero con una reducción notable en la cantidad de capas convolucionales que la conforman. Nuevamente se utiliza al error cuadrático medio (*MSE*) como función de pérdida. La imagen 49 ilustra esta arquitectura:



**Imagen 49.** Esquema PhotoelastNet y componentes. Adaptado de Briñez et al., 2022 [2].

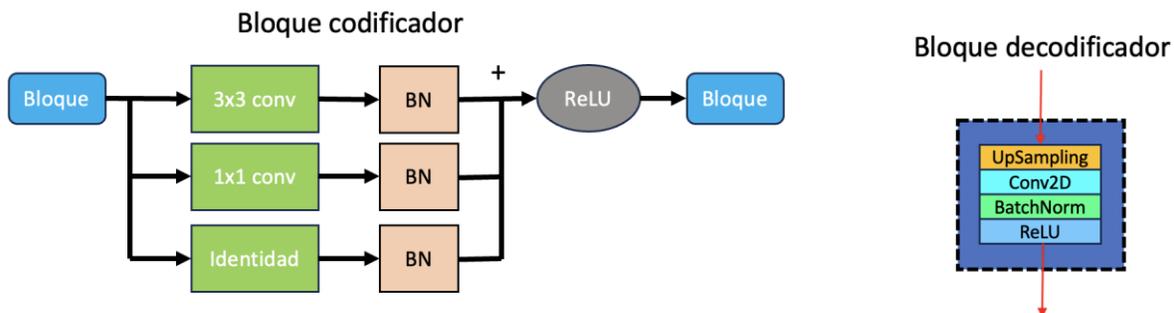
### 5.1.5 VGG – Visual Geometry Group

Algunos meses después de la publicación de la *PhotoelastNet*, Tao et al., 2022 [4] publican la red de recuperación de campos de esfuerzos, en la que no solo se propone una arquitectura diferente a las que se han analizado hasta acá, con una arquitectura VGG, sino que también se optimiza la función de pérdida propuesta en el desarrollo de la *PhotoelastNet*, por medio de una función de optimización objetiva ponderada con múltiples

funciones de pérdida, esta función también será estudiada y evaluada más adelante en este mismo capítulo.

La etapa codificadora de esta arquitectura consiste en una serie de bloques, donde cada uno tiene tres capas: una capa de convolución 3x3, una capa de convolución 1x1 y una capa de identidad. Las dos capas de convolución tienen el objetivo de capturar diferentes niveles de detalles en la imagen de entrada, mientras que la capa de identidad permite que la información original pase directamente sin cambios. Estas capas se concatenan y se aplica una función de activación ReLU para obtener la salida del bloque. El número de filtros en cada bloque va aumentando progresivamente, lo que permite capturar características más complejas a medida que se profundiza en el codificador. Después de cada bloque, se aplica un MaxPooling2D para reducir la resolución espacial de las características.

Por su parte, la etapa decodificadora utiliza bloques de convolución con interpolación bicúbica para realizar un muestreo ascendente (UpSampling2D) de las características. Cada bloque de convolución en el decodificador tiene una estructura similar al codificador, que consiste en una capa de convolución seguida de una capa de *BatchNormalization* y una activación *ReLU*. Estos bloques de convolución ayudan a aumentar la resolución espacial de las características comprimidas y a reconstruir una imagen final. El número de filtros en los bloques de convolución del decodificador disminuye a medida que se acerca a la salida final, lo que permite una progresiva reconstrucción de la imagen. La imagen 50 ilustra las etapas de esta arquitectura:



**Imagen 50.** Bloques que conforman la arquitectura VGG. Adaptado de Tao et al., 2022 [4].

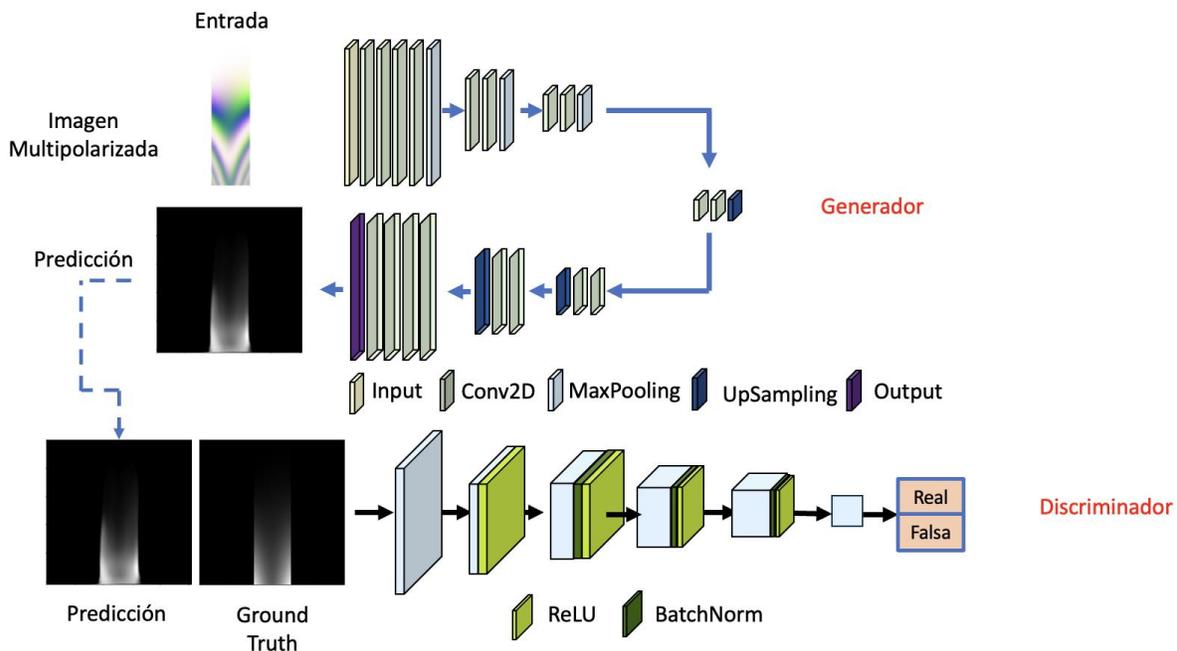
### 5.1.6 GAN + Pix2Pix - UNet

Finalmente, dentro de las arquitecturas consideradas, se analiza la implementación de Park et al., 2022 [46] con una arquitectura GAN + Pix2Pix. Esta arquitectura consta de dos componentes principales: el generador (*generator*) y el discriminador (*discriminator*). Estos dos modelos trabajan en conjunto para generar cada vez imágenes más realistas.

El generador, es el responsable de crear imágenes realistas a partir de una imagen de entrada. En esta arquitectura, el generador replica una arquitectura UNet (*encoder-decoder*) con conexiones de salto. El codificador toma una imagen multipolarizada y la reduce progresivamente a través de capas convolucionales, capturando características a diferentes niveles de abstracción. El decodificador toma la representación latente generada por el codificador y la reconstruye en una imagen de campo de esfuerzos, a la salida.

Por su parte, el discriminador es responsable de distinguir entre imágenes reales y las generadas por el generador. En este caso específico, el discriminador toma como entrada dos imágenes: el campo de esfuerzos producido por el generador y la verdadera imagen objetivo (*Ground Truth*). Ambas imágenes se concatenan y pasan a través de varias capas convolucionales para extraer características y aprender representaciones de las imágenes. El discriminador produce una salida que representa la probabilidad de que la imagen generada sea ‘real’ o ‘falsa’ (un valor entre 0 y 1), basándose en la similitud de ambas imágenes.

La red se entrena en un proceso adversarial, donde el objetivo del generador es ‘engañar’ al discriminador para que clasifique las imágenes generadas como reales. Mientras tanto, el discriminador se entrena para ser capaz de distinguir correctamente entre imágenes reales y generadas. Este proceso de entrenamiento en competencia mutua entre el generador y el discriminador permite mejorar gradualmente la calidad de las imágenes generadas. La imagen 51 ilustra esta arquitectura:



**Imagen 51.** Esquema GAN + Pix2Pix y capas componentes. Adaptado de Park et al., 2022 [46].

## 5.2 Implementación de métricas de calidad de imagen

Las métricas de calidad de imagen son herramientas fundamentales para evaluar el desempeño y la eficacia de las redes neuronales convolucionales (CNN) en tareas de procesamiento y traducción de imágenes. Estas métricas proporcionan una evaluación cuantitativa del rendimiento del modelo en comparación con las imágenes de referencia (*Ground Truth*) o con otros factores de evaluación basados en calificaciones humanas.

En la sección 2.3 de esta tesis se realizó una investigación sobre las métricas de calidad de imagen que habían sido utilizadas en problemas de regresiones de imágenes, desenvolvimiento de fase, imágenes con patrones de franjas, entre otros.

Esta investigación permitió conocer los tipos de métricas existentes, sus fórmulas, orígenes, componentes y su aplicación en este tipo de problemas. A continuación se describe brevemente la información encontrada.

### 5.2.1 Tipos de métricas de calidad de imagen

Tal como se menciona en la sección 2.3 del estado del arte, las métricas de calidad de imagen se dividen en dos categorías: Métricas “No reference” (NR) y métricas “Full Reference” (FR).

Las NR son aquellas capaces de otorgar una calificación a una imagen sin requerir de otra imagen de referencia con la cual comparar, ya que generalmente utilizan algoritmos pre-entrenados para brindar una valoración a una imagen teniendo en cuenta sus características. Es el caso de las métricas BRISQUE, DBCNN, NIMA, NRQM, CKDN, entre otras.

Por su parte, las métricas FR corresponden a aquellas en las que se cuenta con una imagen de referencia con la cual comparar una imagen obtenida por una red neuronal. Es el caso de las métricas DISTS y LPIPS, MSE, SSIM, PSNR, entre otras.

En esta tesis, se decide utilizar las métricas FR para comparar las imágenes producidas por las diferentes arquitecturas con su respectivo *ground truth* y con ello, realizar una evaluación de las predicciones de cada una de las arquitecturas. Por su parte, también se utilizarán las métricas NR sobre la *MultipolarNet*, con la finalidad de evaluar sus predicciones desde diferentes perspectivas que vienen implícitas en estas métricas.

### 5.2.2 Métricas tradicionales

Siguiendo con la revisión realizada en la sección 2.3 del estado del arte, se evidenció que muchos de los estudios antecedentes a esta tesis utilizaban las métricas MSE, SSIM y PSNR para evaluar el desempeño de los modelos propuestos.

Estas tres métricas son utilizadas en esta tesis para la evaluación de los diferentes modelos de red neuronal generados y también para la construcción de funciones de pérdida simples y compuestas, con las que se optimizan las predicciones de las redes.

Estas tres métricas existen actualmente y son asequibles mediante código abierto en librerías como `skimage.metrics`.

### 5.2.3 Librería pyiqa

Por su parte, se realiza también una búsqueda que permita implementar otras métricas NR y FR con la finalidad de realizar más mediciones sobre la *MultipolarNet*, en la sección 5.7.

Se encontró que este tipo de métricas están disponibles en la librería de código abierto *pyiqa* (*PyTorch Toolbox for Image Quality Assessment*). Por medio de esta librería es posible importar métricas como BRISQUE, DBCNN, NIMA, NRQM, CKDN, DISTS y LPIPS.

#### 5.2.4 Métricas adicionales

Adicional a las métricas de calidad de imagen tradicionales y a las encontradas en la librería *pyiqa*, se encontraron otras dos métricas de calidad de imagen FR propias, que se utilizan dentro de los refinamientos en las funciones de pérdida, en la sección 5.5. Estas métricas son:

**Métrica de la Diferencia Máxima Absoluta (MAD):** Mide la mayor diferencia absoluta entre el valor de un píxel en la imagen original (*ground truth*) y su correspondiente píxel en la imagen producida por la red. Para calcular su valor se debe:

1. Calcular la diferencia absoluta entre cada píxel de la imagen original (*ground truth*) y su correspondiente píxel en la imagen producida por la red.
2. Encontrar el valor máximo de la diferencia absoluta obtenida en el paso anterior.

El resultado final es el valor MAD, que representa la mayor diferencia absoluta entre los valores de los píxeles de ambas imágenes.

**Error de píxeles brillantes (BPE).** Evalúa la calidad de las imágenes producidas por una red en términos de los valores máximos de los píxeles. Cuanto menor sea el valor del BPE, mejor será la calidad de la imagen en términos de la conservación de los valores máximos de los píxeles. Para calcular su valor se debe:

1. Calcular el valor máximo de los píxeles en la imagen original (*ground truth*) y en la imagen producida por la red.
2. Calcular la diferencia entre los valores máximos de los píxeles de ambas imágenes.

### 5.3 Implementación de arquitecturas

Tomando el conocimiento obtenido de cada una de las arquitecturas propuestas en la sección 5.1, se intentó desarrollar cada una de ellas utilizando las librerías *Keras* y *Tensorflow*, en *Python*. Se buscó mantener su estructura de manera fiel a la arquitectura original declarada por los autores. No se incluyen capas de *Dropout* ni regularización. La adición y el refinamiento de las capas de regularización se realizará por separado sobre la arquitectura que presente mejores métricas. El código de cada una de estas arquitecturas puede encontrarse en el Anexo 3.

Las arquitecturas implementadas y analizadas en esta tesis presentan variedad en cuanto a su composición, tamaño e hiper-parámetros, es por ello que se hace relevante visualizar en un primer momento sus características generales, para también tener esto en cuenta a la hora de evaluar las predicciones efectuadas por cada red.

En este caso inicial, todas las arquitecturas serán entrenadas con *MSE* como función de pérdida y con *Adam* como optimizador, con la finalidad de realizar una selección basándose únicamente en la capacidad de predicción de cada red por su composición. La tabla 1 resume las características de los modelos luego de su implementación en *Python*.

Tabla 1. Características generales de arquitecturas consideradas.

Arquitectura	Autor	Parámetros entrenables	Parám No Entrenables	Capas Conv2D	Número Min Filtros	Número Max Filtros	Función de Costo	Optimizador
UNet	Ronnenberg et al., 2015 [41]	31.391.297	11.776	18	64	1.024	MSE	Adam
Unet + LSTM	Perera et al., 2020 [14]	895.905	992	15	16	128		
StressNet	Briñez et al., 2020 [3]	45.969.985	23.040	33	64	512		
PhotoelastNet	Briñez et al., 2022 [2]	972.949	1.664	19	4	256		
VGG	Tao et al., 2022 [4]	45.341.187	9.858	15	64	1.024		
GAN + Pix2Pix	Park et al., 2022 [46]	51.661.889	9.856	16	64	512		

A partir de estas arquitecturas, se realiza un entrenamiento de distintos modelos utilizando la colección de imágenes desarrolladas en el capítulo 4, con la finalidad de encontrar una arquitectura base, la cual alcanzara los mejores valores dentro de las métricas de calidad de imagen tradicionales (MSE, SSIM y PSNR) en las colecciones de validación.

Este entrenamiento se realiza por lotes de 2500 imágenes de entrenamiento (80%) y 625 imágenes de validación (20%), por cuatro iteraciones, hasta completar 10000 imágenes de entrenamiento y 2500 imágenes de validación. Como la idea de esta primera búsqueda es únicamente evaluar el rendimiento de las arquitecturas, se decide entrenarlas a todas utilizando una misma configuración general: utilizando MSE como función de pérdida y Adam como optimizador (su selección y optimización se realiza más adelante).

Cada modelo se entrena basado en la última versión de su modelo predecesor. Esto es, un modelo entrenado con las imágenes 2501 a 5000 toma los pesos del modelo que ya fue entrenado previamente con las imágenes 1 a 2500 y continúa con su entrenamiento.

Se utiliza un *BatchSize* de 4. En cada iteración se configura un máximo de 100 épocas. Sin embargo, se superponen dos callbacks para evitar sobre-entrenamiento: un *EarlyStopping* y un *ModelCheckpoint*. El *EarlyStopping* se configura con una paciencia de 5, lo cual finaliza el entrenamiento en caso de que la función de pérdida en la colección de validación (*val\_loss*) no mejore durante 5 épocas, es decir, no disminuya el MSE. Por su parte, el *ModelCheckpoint* es el encargado de guardar en localmente únicamente al modelo que logre vencer el mejor valor de la función de pérdida obtenida hasta ese momento, es decir, al modelo de menor MSE.

En esta sección se muestran gráficamente las predicciones obtenidas por cada uno de los modelos desarrollados. La selección de la arquitectura base, conforme a las métricas de calidad de imagen se realiza en la sección 5.4.

Inicialmente, se utilizan los modelos entrenados para realizar predicciones sobre el conjunto de imágenes de validación (modelos de disco, anillo, placa y vigas). A continuación se ilustran las imágenes obtenidas por cada uno de los modelos (Imágenes 52 a 55):

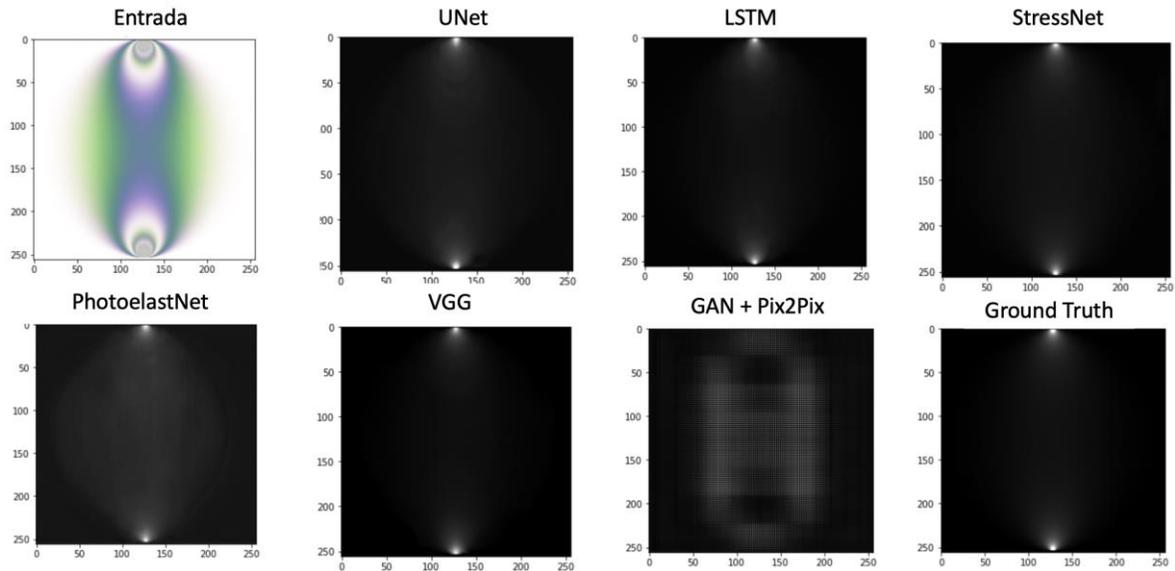


Imagen 52. Resultados sobre geometría de disco a compresión para las arquitecturas estudiadas.

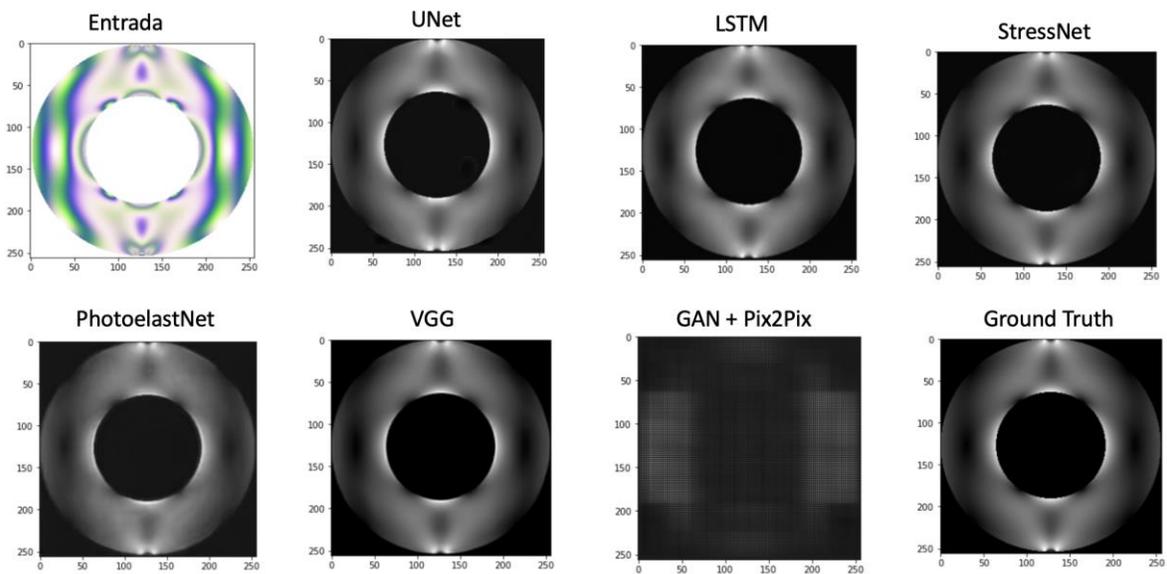


Imagen 53. Resultados sobre geometría de anillo a compresión para las arquitecturas estudiadas.

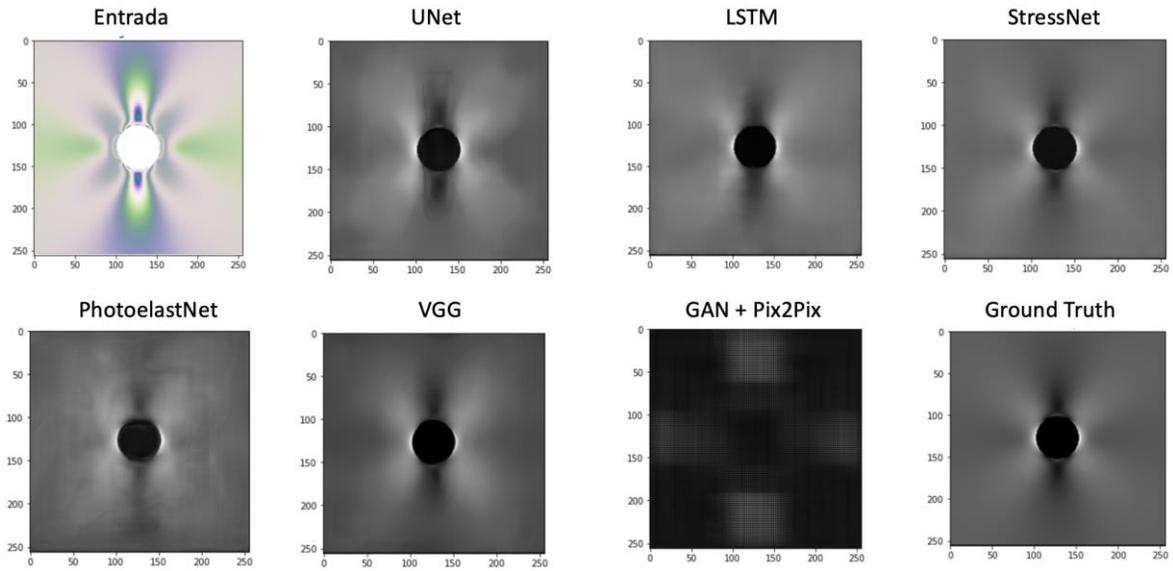


Imagen 54. Resultados sobre geometría de placa infinita para las arquitecturas estudiadas.

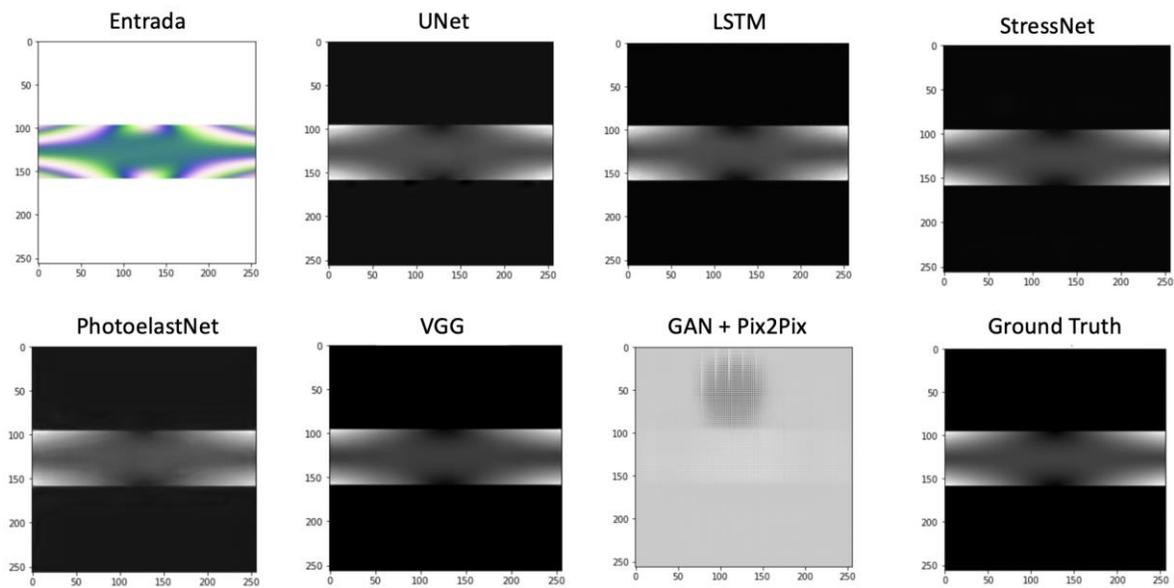
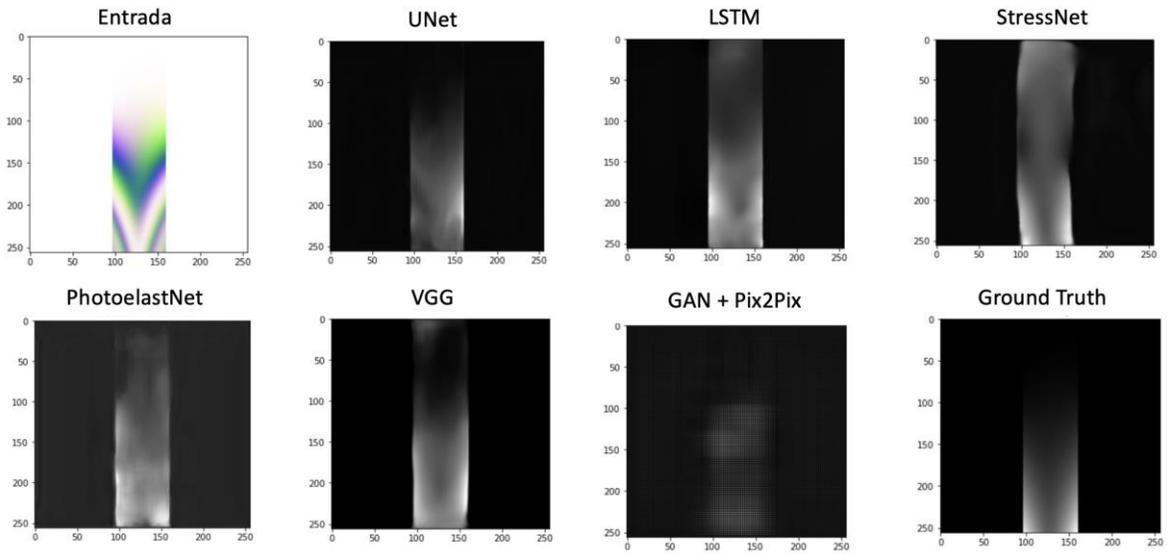
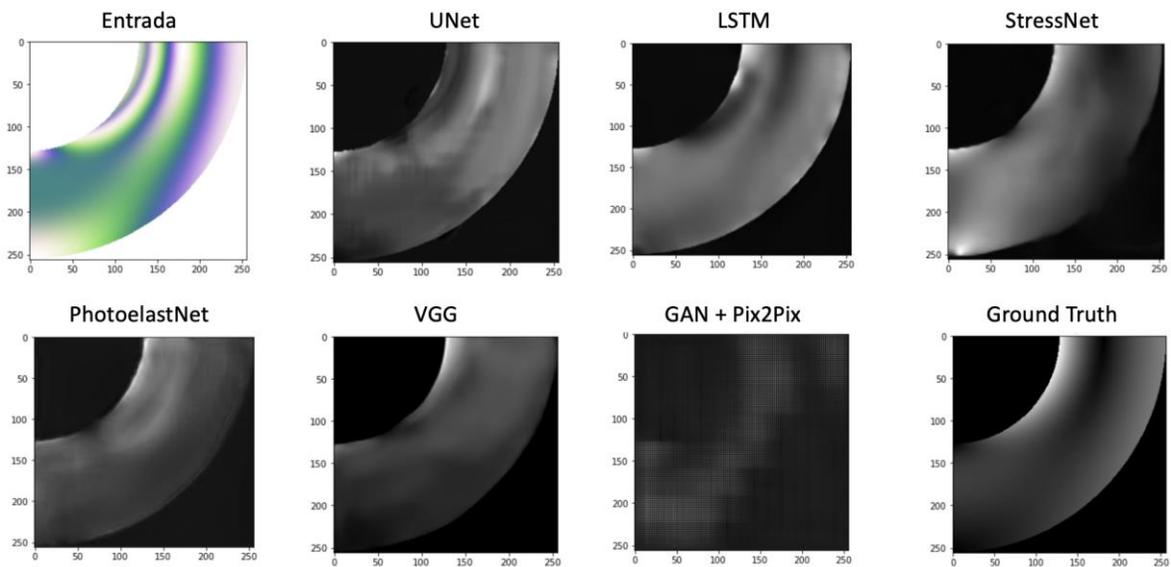


Imagen 55. Resultados sobre geometría de viga bajo carga puntual para las arquitecturas estudiadas.

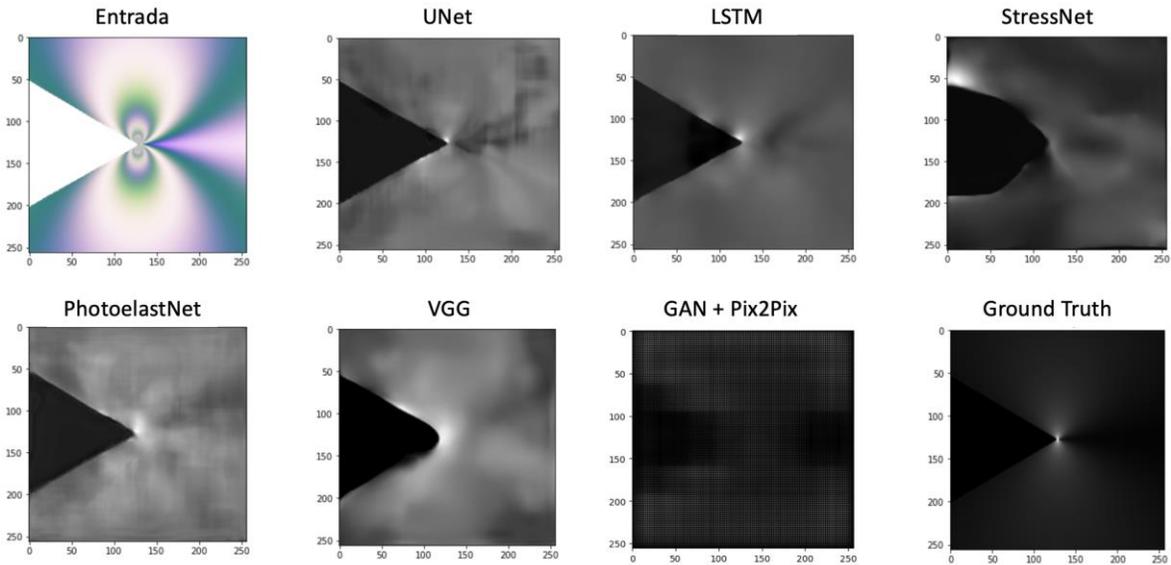
De igual manera, se obtuvieron resultados sobre las geometrías desconocidas para las redes, aquellas que no fueron utilizadas en la fase de entrenamiento. Las imágenes 56 a 58 ilustran estos resultados:



**Imagen 56.** Resultados sobre geometría de estructura vertical bajo presión lateral para las arquitecturas estudiadas.



**Imagen 57.** Resultados sobre geometría de ménsula curva cargada en su extremo libre para las arquitecturas estudiadas.



**Imagen 58.** Resultados sobre geometría de placa rectangular finita con muesca tipo 'V' bajo compresión uniaxial para las arquitecturas estudiadas.

A través de los resultados, es posible evidenciar buenos resultados en las geometrías utilizadas durante el entrenamiento, lo cual es un indicio de que las redes son capaces de predecir campos de esfuerzos ante diferentes condiciones en las imágenes, mientras las geometrías permanezcan conocidas.

Al utilizar geometrías desconocidas, los modelos se alejan un poco más y las predicciones ya no son tan precisas, lo cual sugiere que los modelos entrenados aún no son versátiles y no cuentan con la capacidad suficiente de predecir ante geometrías desconocidas. Esto, sin embargo, es algo esperable, teniendo en cuenta que a este punto no se ha realizado ningún proceso de refinamiento interno de las redes neuronales, es decir, no se han agregado capas de regularización o *Dropout* para evitar sobre-entrenamientos y mejorar la capacidad de predicción ante geometrías desconocidas. Tampoco se han optimizado más componentes de la red, como la función de pérdida, las capas ocultas y algunos otros hiperparámetros, que se refinarán más adelante en este mismo capítulo.

El caso de la arquitectura GAN + Pix2Pix es un caso especial que sugiere que quizá hacían falta un poco más de iteraciones durante el entrenamiento, pues si bien es el resultado que más se aleja a la predicción esperada en todas las geometrías, es posible observar que las predicciones que genera tienen la forma inicial esperada, por lo que, quizá con un mayor entrenamiento y refinamiento, los resultados podrían mejorar notablemente.

En cuanto a las otras arquitecturas estudiadas, analizando únicamente las imágenes resultantes, se podría afirmar que, en general, la predicción es buena, pero se requiere conocer las métricas de calidad de imagen para poder llevar a cabo una selección de la arquitectura base que se seguirá refinando cada vez más durante todo este capítulo.

Estas métricas se muestran y analizan a continuación, en la sección 5.4.

## 5.4 Selección de una arquitectura base

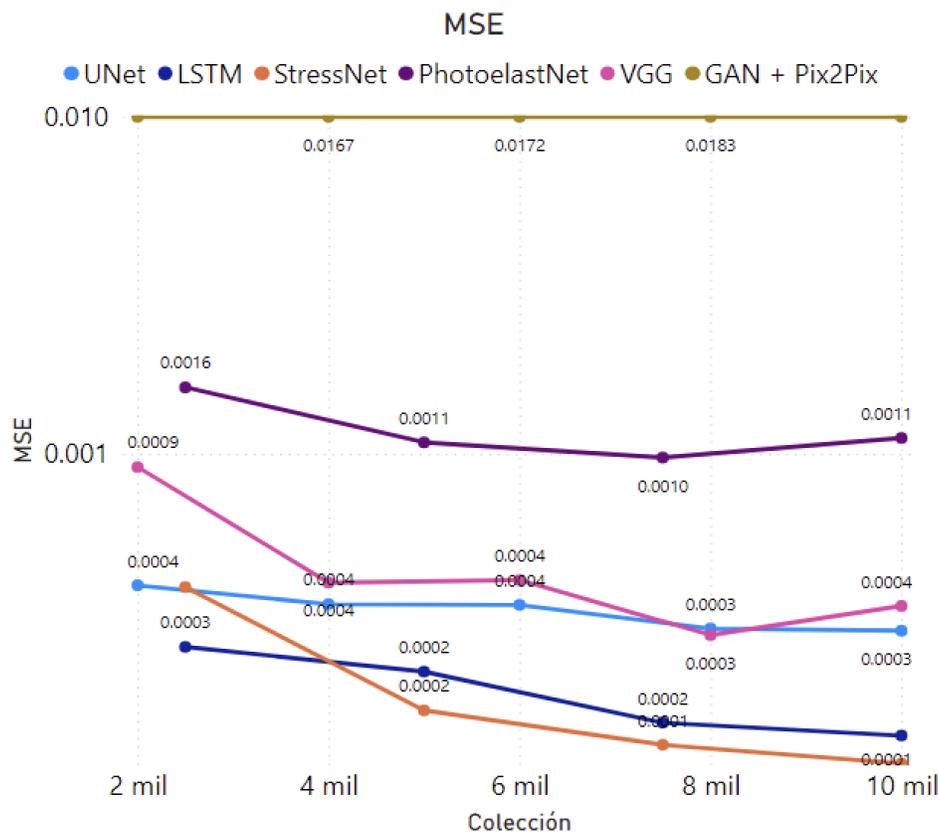
No basta únicamente con conocer los resultados gráficos de las predicciones realizadas por las diferentes arquitecturas sobre las geometrías multipolarizadas, sino que se hace necesario conocer numéricamente el valor generado sobre cada una de las métricas de calidad de imagen por cada una de las arquitecturas consideradas.

Para lograrlo, se toman las arquitecturas estudiadas y se evalúan las predicciones realizadas sobre las geometrías multipolarizadas, tanto las conocidas como las desconocidas.

### 5.4.1 Resultados sobre geometrías conocidas

Las imágenes 59 a 61 contienen los resultados gráficos de las diferentes arquitecturas en las métricas tradicionales MSE, SSIM y PSNR, para cada uno de los ciclos de entrenamiento y validación con las geometrías conocidas por las redes (disco, anillo, placa y viga).

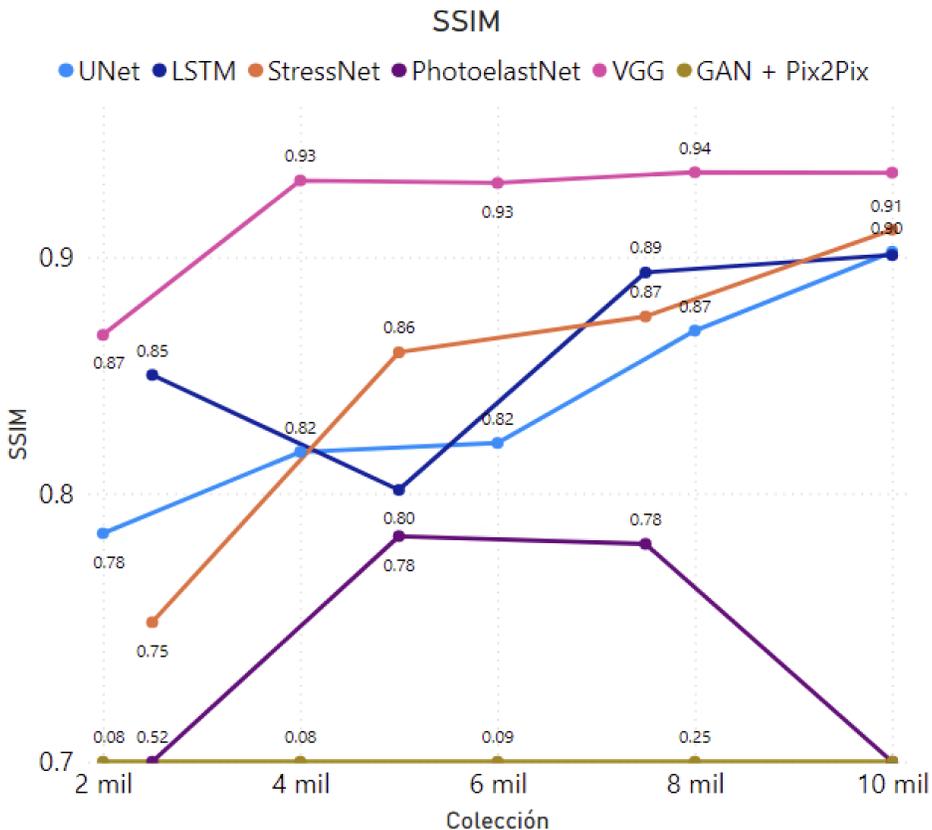
Comenzando por el MSE, de la imagen 59 se puede observar que las arquitecturas StressNet y LSTM presentan, en promedio, un menor error cuadrático medio entre los píxeles de las imágenes generadas por estas redes y el *ground truth*:



**Imagen 59.** Métrica del error cuadrático medio (MSE) para la colección de geometrías conocidas, por arquitectura.

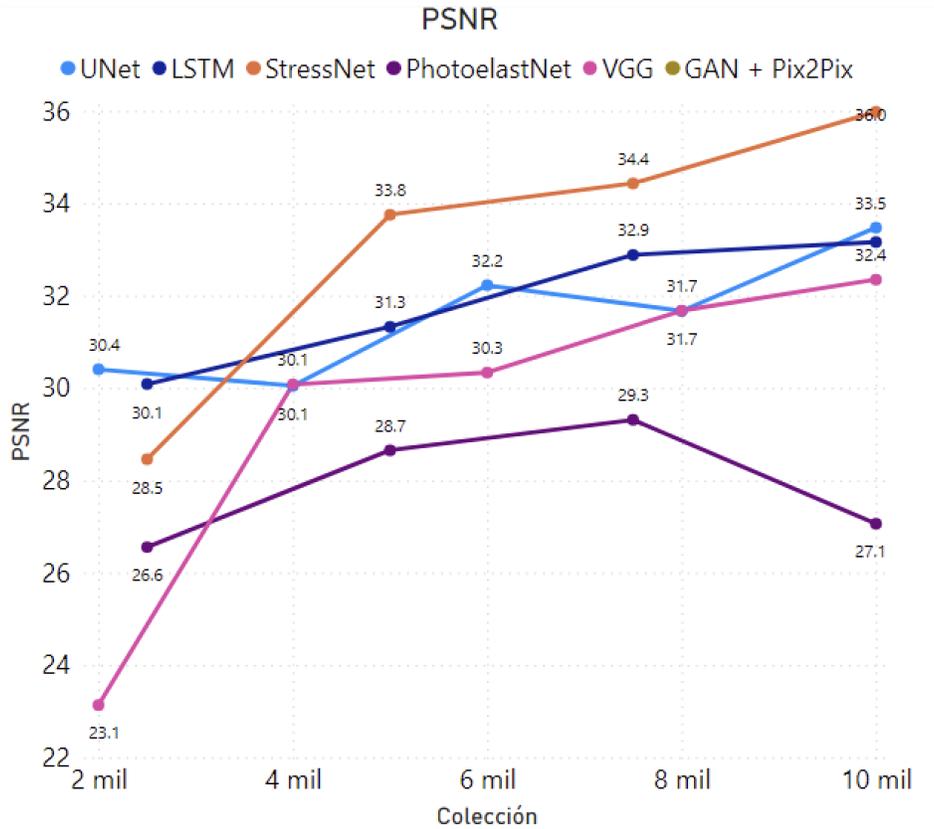
Por su parte, de la imagen 60 se puede observar que la arquitectura VGG presenta, en promedio, el valor más elevado para la métrica del índice de similitud estructural (SSIM), es decir, que las imágenes generadas por esta arquitectura son estructuralmente más semejantes al *ground truth* que las de las otras arquitecturas consideradas.

Las arquitecturas StressNet, LSTM y UNet también presentan altos valores con una tendencia creciente, por detrás de la arquitectura VGG.



**Imagen 60.** Métrica del Índice de similitud estructural (SSIM) para la colección de geometrías conocidas, por arquitectura.

Finalmente, de la imagen 61 se puede observar que la arquitectura StressNet presenta, en promedio, el valor más elevado para la métrica de la proporción máxima de señal a ruido (PSNR), es decir, que la calidad de las imágenes generadas por esta es alta y que la distorsión o pérdida de detalles es baja, en comparación con las de las otras arquitecturas consideradas.

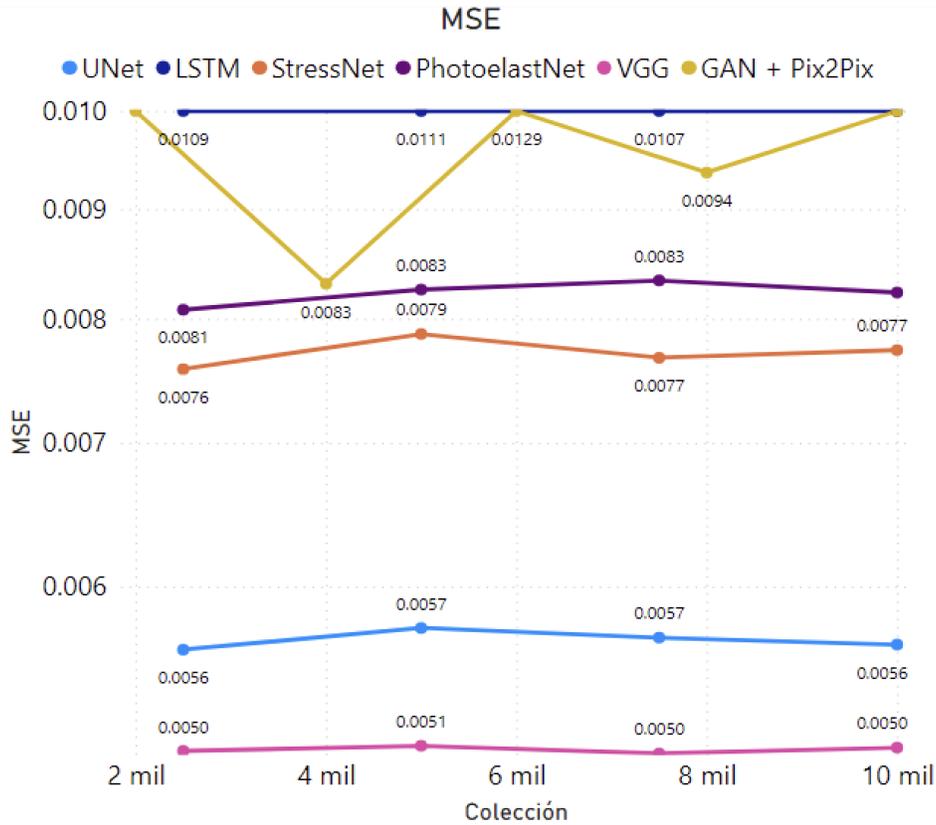


**Imagen 61.** Métrica de la Proporción máxima de señal a ruido (PSNR) para la colección de geometrías conocidas, por arquitectura.

#### 5.4.2 Resultados sobre geometrías desconocidas

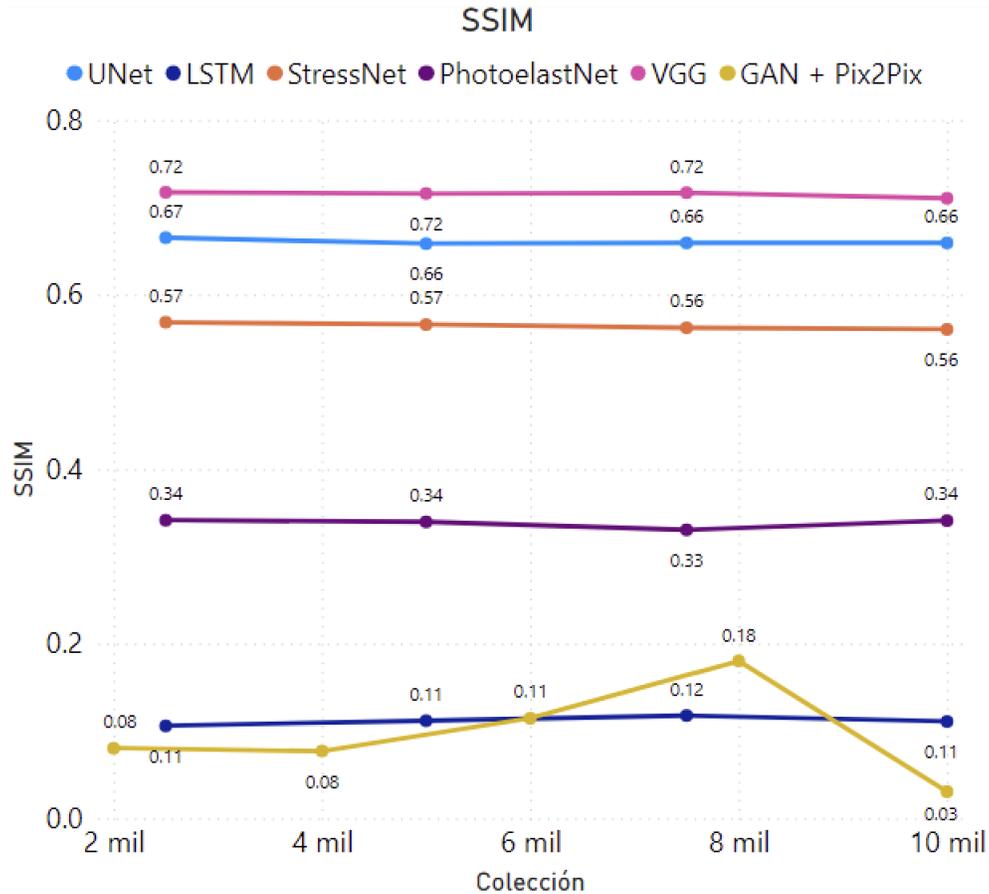
De manera similar, se calcularon las métricas sobre la colección de imágenes desconocidas que nunca fueron utilizadas en el entrenamiento de las redes. Las imágenes 62 a 64 ilustran los resultados obtenidos en estas geometrías.

Nuevamente comenzando con el error cuadrático medio, se obtiene el menor valor de error promedio dentro de las geometrías desconocidas en la arquitectura VGG, seguida por la arquitectura UNet, tal como se muestra a continuación, en la imagen 62:



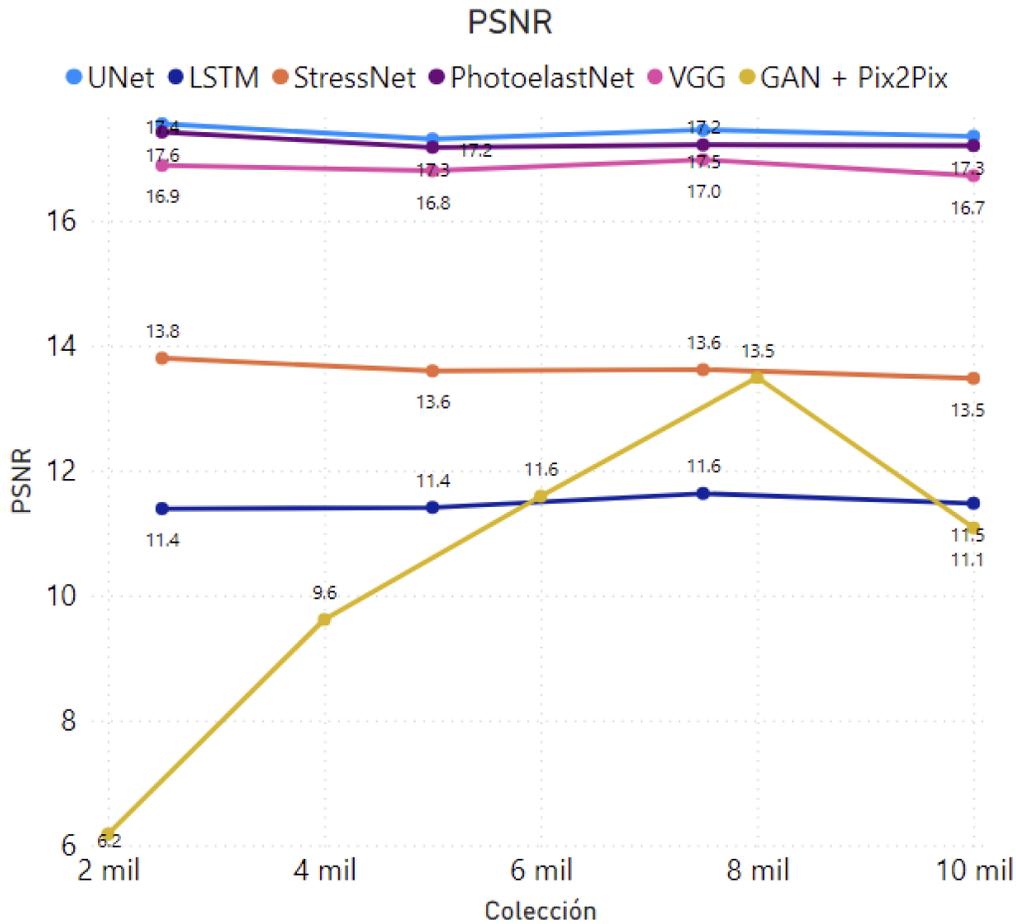
**Imagen 62.** Métrica del error cuadrático medio (MSE) para la colección de geometrías desconocidas, por arquitectura.

Por su parte, la métrica SSIM presenta, en promedio, sus mejores valores con las geometrías desconocidas también en la arquitectura VGG, nuevamente seguida por la arquitectura UNet, tal como se muestra a continuación, en la imagen 63:



**Imagen 63.** Métrica del Índice de similitud estructural (SSIM) para la colección de geometrías desconocidas, por arquitectura.

Finalmente, en la imagen 64 se puede observar que las arquitecturas UNet y PhotoelastNet presentan, en promedio, el valor más elevado para la métrica de la proporción máxima de señal a ruido (PSNR) utilizando la colección de imágenes desconocidas. Siguiendo muy de cerca a estas dos arquitecturas, también se encuentra presente la arquitectura VGG.



**Imagen 64.** Métrica de la Proporción máxima de señal a ruido (PSNR) para la colección de geometrías desconocidas, por arquitectura.

A modo de comparación, también es importante resaltar las diferencias obtenidas en las métricas de calidad de imagen al utilizar la colección de geometrías conocidas y al utilizar la colección de geometrías desconocidas: En las tres métricas (MSE, SSIM y PSNR) se obtienen mejores valores para la colección conocida, lo cual sugiere una necesidad de implementar métodos que eviten el sobre-entrenamiento de las redes y mejoren su capacidad de predicción ante geometrías desconocidas.

### 5.4.3 Selección

Basándose en los resultados previamente mostrados sería posible encontrar un predominio en la calidad de imágenes conocidas en la geometría StressNet, al tener las mejores métricas MSE y PSNR. Siguiendo a la StressNet, podría hablarse de las arquitecturas VGG y LSTM. La VGG obtiene la métrica SSIM más alta y la LSTM tiene buenas métricas en general.

Por su parte, dentro de la colección de imágenes desconocidas, la arquitectura VGG sobresale al tener las mejores métricas MSE y SSIM, seguida de la arquitectura UNet, con el mejor valor en la métrica PSNR y buenas métricas generales.

Teniendo esto en cuenta, se debe realizar una selección entre estas arquitecturas para continuar con la selección de la función de pérdida (sección 5.5) y la optimización de las componentes de la red (sección 5.6).

Como factor de selección, se decide dar una mayor importancia a los resultados obtenidos sobre imágenes desconocidas, pues el objetivo a lograr con la red es predecir sobre geometrías complejas, en muchos casos desconocidas, ante diversas condiciones de iluminación, cámaras, ruido, entre otros.

Siendo así, se determina que la arquitectura base es la arquitectura VGG, debido a los resultados obtenidos en las métricas MSE y SSIM en las imágenes desconocidas.

## 5.5 Selección de la función de pérdida

Una vez se ha realizado una selección de la arquitectura base, fundamentada en los resultados mostrados, se debe continuar con la construcción de un modelo refinado de red neuronal. En esta sección se aborda la selección de la función de pérdida partiendo de algunas funciones planteadas por otros autores y también algunas más propuestas dentro del desarrollo de esta tesis. Posteriormente se entrena la arquitectura VGG con cada una de las funciones de pérdida propuestas, se mantienen los demás hiper-parámetros fijos y se evalúan los resultados obtenidos con cada una de las funciones de pérdida.

### 5.5.1 Funciones de pérdida basadas en otros autores

#### **Error cuadrático medio MSE:**

Dentro de las funciones de pérdida consideradas, se decide tener en cuenta al tradicional error cuadrático medio, una de las métricas más básicas pero a su vez más utilizadas como función de pérdida en el entrenamiento de redes neuronales.

#### **Métrica del Índice de Similitud Estructural SSIM:**

De igual forma a como se realiza con el MSE, también se tiene en cuenta a la métrica SSIM como función de pérdida, dada su importancia en la medición de la similitud entre las imágenes producidas por las redes y el *ground truth*.

#### **Función de pérdida de Perera et al., 2021 [14]:**

En su artículo, estos autores demuestran que utilizar al clásico MSE como función de pérdida muestra una convergencia insuficiente que da lugar a un rendimiento de desenvolvura de fase deficiente y que el problema de desenvolvimiento de fase de no tiene una solución única.

Dado que el MSE obliga a la red a aprender una solución única, no se ajusta bien a la naturaleza del problema de desenvolvura de fase. Por lo tanto, se requiere una función de pérdida que permita otras soluciones en la convergencia al tiempo que aumenta la similitud entre la fase predicha  $\hat{\phi}$  y la fase verdadera  $\phi$ .

Para resolver estos problemas, Perera adopta la función de pérdida compuesta  $\mathcal{L}_c$  definida a continuación (ecuaciones 48 a 50):

$$\mathcal{L}_c = \lambda_1 \mathcal{L}_{var} + \lambda_2 \mathcal{L}_{tv} \quad (48)$$

Donde:

$$\mathcal{L}_{var} = \mathbb{E} \left[ (\hat{\phi} - \phi)^2 \right] - (\mathbb{E}[(\hat{\phi} - \phi)])^2 \quad (49)$$

$$\mathcal{L}_{tv} = \mathbb{E} [ |\hat{\phi}_x - \phi_x| + |\hat{\phi}_y - \phi_y| ] \quad (50)$$

Y  $\lambda_1 = 1$ ,  $\lambda_2 = 0.1$  son los pesos asignados para las dos pérdidas, fijados empíricamente durante el entrenamiento.

#### **Función de pérdida de Tao et al 2022 [4]**

Siguiendo con la visión de buscar una alternativa contra la convergencia rápida y errónea del MSE, estos autores propusieron una función de pérdida basándose en los errores denominados  $L1$  y  $L2$ , en agrupación con la métrica de calidad de imagen SSIM. Las ecuaciones 51 a 53 explican esta función de pérdida:

$$\mathcal{L} = \alpha \mathcal{L}^{MS_{SSIM}}(I_r, I_g) + \beta \mathcal{L}^{L1}(I_r, I_g) + (1 - \alpha - \beta) \mathcal{L}^{L2}(I_r, I_g) \quad (51)$$

Donde:

$$\mathcal{L}^{L1}(I_r, I_g) = \frac{1}{N} \sum_{p \in I} |I_r(p) - I_g(p)| \quad (52)$$

$$\mathcal{L}^{L2}(I_r, I_g) = \frac{1}{N} \sum_{p \in I} [I_r(p) - I_g(p)]^2 \quad (53)$$

Y  $\mathcal{L}^{MS_{SSIM}}$  corresponde a la métrica SSIM,  $p$  es el índice del píxel,  $I_r(p)$  e  $I_g(p)$  son los valores de los píxeles en el campo de esfuerzos construido por la red y en el *ground truth*, respectivamente y  $N$  es el número de píxeles  $p$  en la imagen  $I$ . Según los autores,  $\alpha = 0.5$  y  $\beta = 0.25$ .

### **5.5.2 Funciones de pérdida propias**

#### **Función de pérdida a partir de los errores MSE, MAE y la métrica SSIM ponderados ( $\mathcal{LF}$ )**

Consiste en una función de pérdida que involucra dos de los errores más típicos utilizados en métricas de calidad de imagen: el error cuadrático medio MSE y el error absoluto medio MAE, además de incluir a la métrica SSIM en forma de promedio (pesos iguales), teniendo en cuenta que tanto el MSE como el MAE se optimizan hacia el mínimo, mientras que la métrica SSIM lo hace hacia el máximo. La ecuación 54 representa esta función de pérdida propuesta:

$$\mathcal{L}\mathcal{F} = \frac{1}{3} [MSE + MAE + (1 - SSIM)] \quad (54)$$

Donde:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \tilde{y}_i| \quad (55)$$

Y MSE y SSIM siguen las ecuaciones (7) y (9) respectivamente. El objetivo de esta función de pérdida será minimizarla, para reducir los valores del MAE y del MSE, y a su vez maximizar la métrica SSIM.

### **Función de pérdida a partir de las métricas MAD, BPE y SSIM ( $\mathcal{L}\mathcal{F}_2$ )**

Consiste en una suma aritmética de los valores de estas tres métricas, teniendo en cuenta que tanto el MAD como el BPE se optimizan hacia el mínimo, mientras que la métrica SSIM lo hace hacia el máximo. La ecuación 56 representa esta función de pérdida propuesta:

$$\mathcal{L}\mathcal{F}_2 = MAD + BPE + (1 - SSIM) \quad (56)$$

El objetivo de esta función de pérdida será minimizarla, para reducir los valores del MAD y del BPE, y a su vez maximizar la métrica SSIM.

### **Función de pérdida a partir de los errores MSE, MAE y las métricas MAD, BPE y SSIM ( $\mathcal{L}\mathcal{F}_3$ )**

Consiste en una suma aritmética de los valores de estos errores y métricas, teniendo en cuenta que MSE, MAE, MAD y BPE se optimizan hacia el mínimo, mientras que la métrica SSIM lo hace hacia el máximo. La ecuación 57 representa esta función de pérdida propuesta:

$$\mathcal{L}\mathcal{F}_3 = MSE + MAE + MAD + BPE + (1 - SSIM) \quad (57)$$

El objetivo de esta función de pérdida será minimizarla, para reducir los valores del MSE, MAE, MAD y BPE, y a su vez maximizar la métrica SSIM.

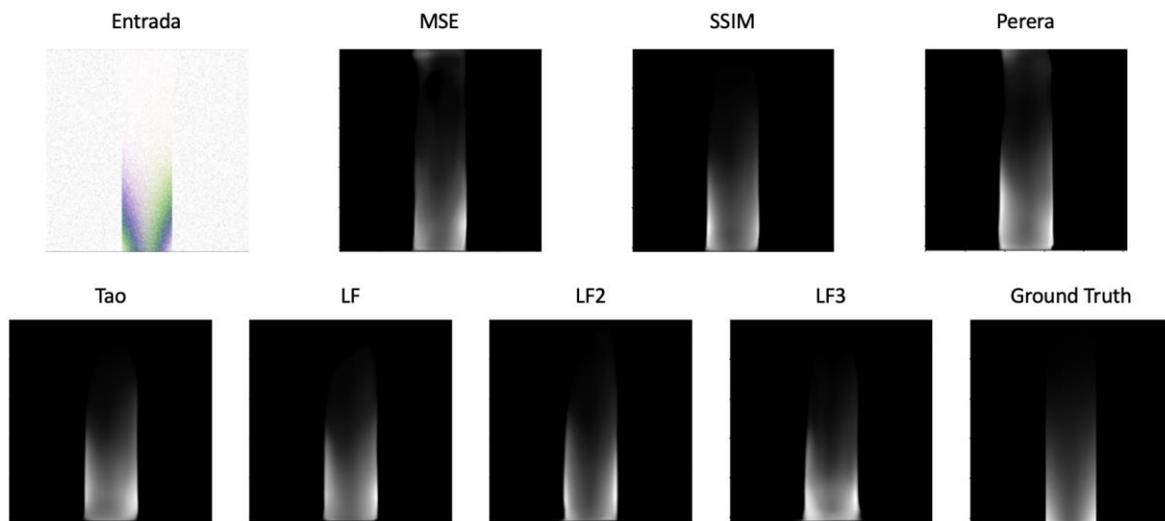
## **5.5.3 Entrenamiento de modelos y resultados**

Partiendo de las siete funciones de pérdida propuestas, se realiza un experimento para determinar cuál de ellas alcanza las mejores métricas de calidad de imagen, si se mantienen los demás factores constantes.

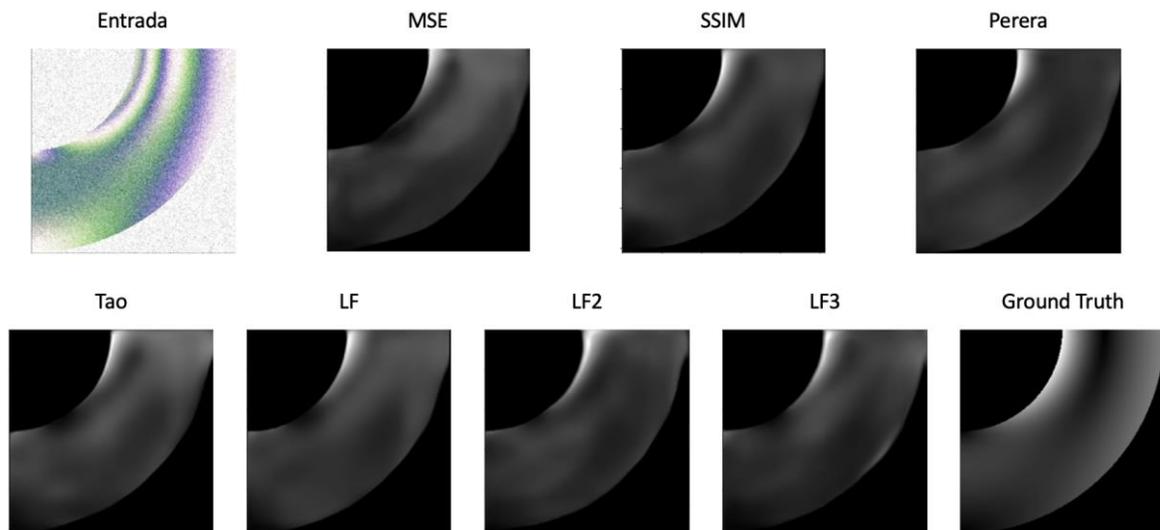
En todos los modelos se utiliza la arquitectura VGG, pues fue la seleccionada en la sección 5.4.3. Se realiza un entrenamiento similar al que implementado en el experimento para la selección de la arquitectura base, esto es, por lotes de 2500 imágenes de entrenamiento (80%) y 625 imágenes de validación (20%), por cuatro iteraciones, hasta completar 10000 imágenes de entrenamiento y 2500 imágenes de validación. Se utiliza Adam como optimizador, *BatchSize* de 4, un máximo de 100 épocas y callbacks de *EarlyStopping* y *ModelCheckpoint*. para evitar sobre-entrenamiento.

En este caso, se evalúan los modelos únicamente sobre las geometrías desconocidas, pues el objetivo es alcanzar las mejores métricas sobre imágenes que no fueron utilizadas en el entrenamiento. A continuación se ilustran las imágenes obtenidas por la arquitectura VGG

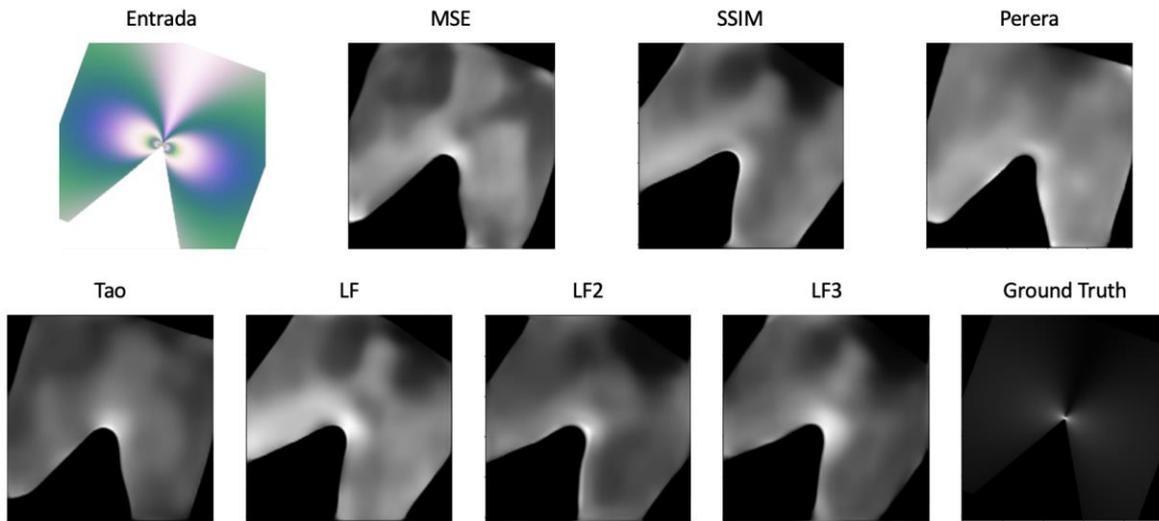
para cada una de las funciones de pérdida propuestas en las geometrías desconocidas (Imágenes 65 a 67):



**Imagen 65.** Resultados sobre geometría de estructura vertical bajo presión lateral para las funciones de pérdida estudiadas.



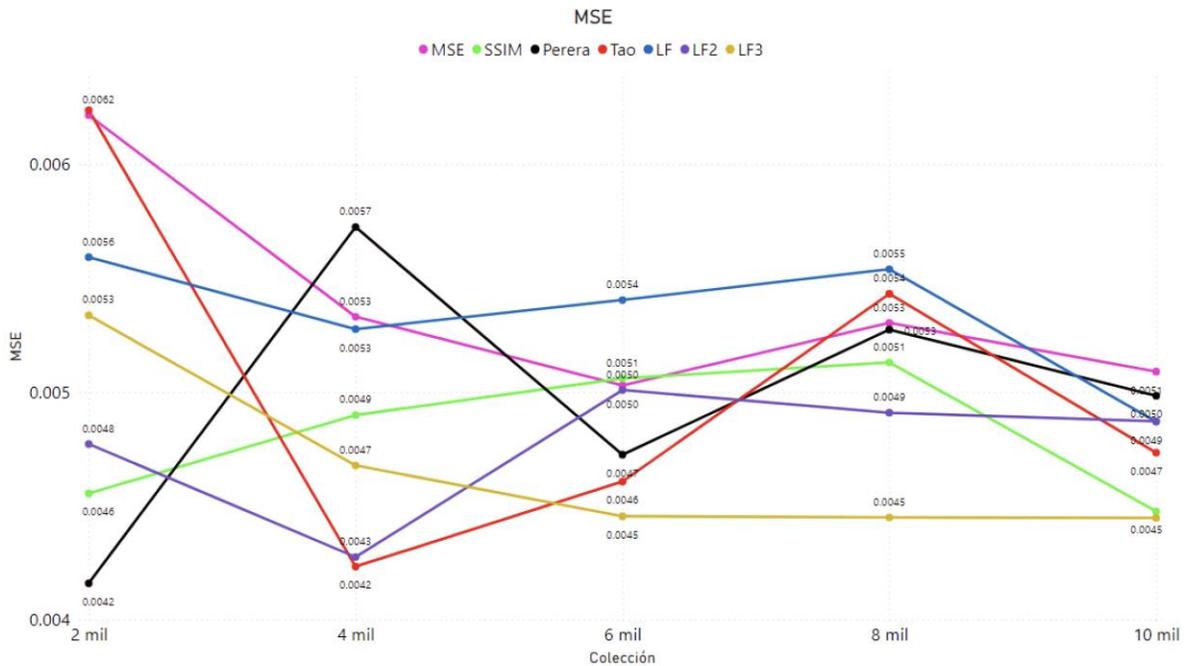
**Imagen 66.** Resultados sobre geometría de ménsula curva cargada en su extremo libre para las funciones de pérdida estudiadas.



**Imagen 67.** Resultados sobre geometría de placa rectangular finita con muesca tipo 'V' bajo compresión uniaxial para las funciones de pérdida estudiadas.

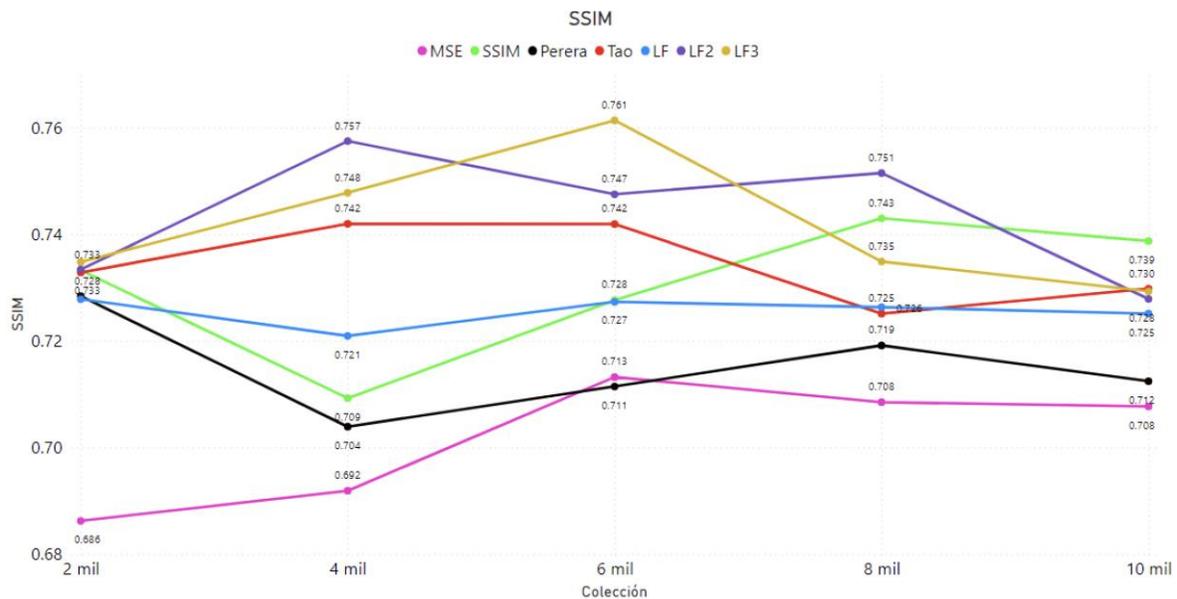
Y de igual manera se grafican los resultados de las métricas de calidad de imagen para cada una de las funciones de pérdida propuestas. Las imágenes 68 a 70 contienen los resultados gráficos de las diferentes funciones de pérdida en las métricas tradicionales MSE, SSIM y PSNR, para cada uno de los ciclos de entrenamiento y validación con las geometrías desconocidas por las redes (estructura vertical, ménsula y placa con muesca).

En la imagen 68 se puede observar que las funciones de pérdida propuestas por los autores Perera y Tao alcanzan los valores mínimos globales para la métrica MSE, pero también vale la pena resaltar el desempeño mostrado por la función de pérdida  $\mathcal{LF}_3$ , la cual muestra una tendencia decreciente constante y alcanza valores muy cercanos a los de las funciones de pérdida planteadas por los dos autores.



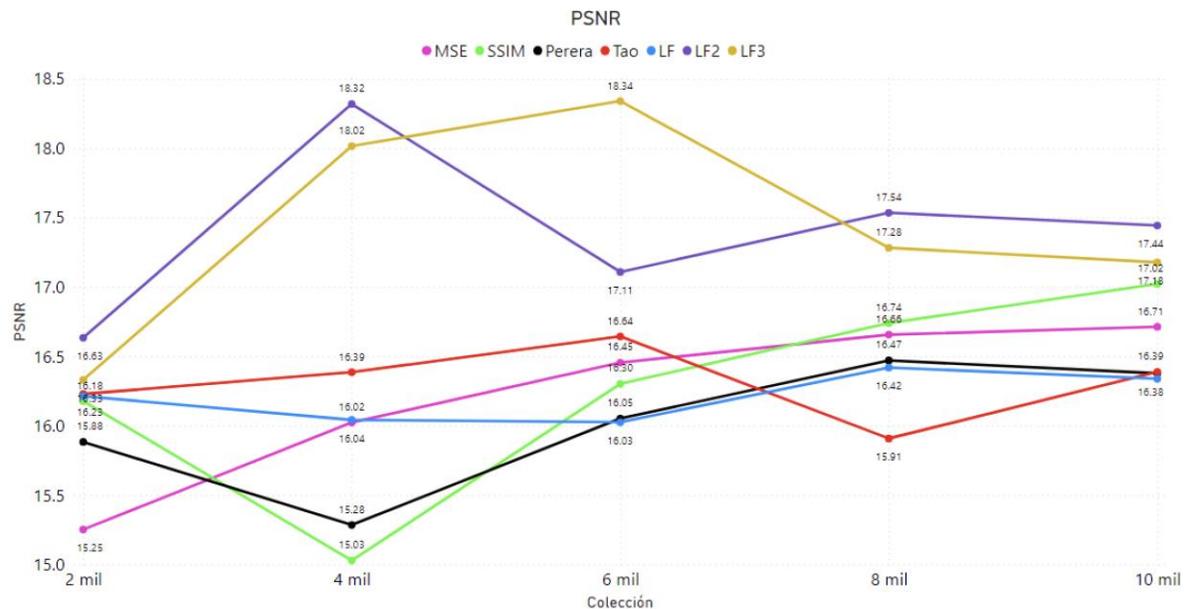
**Imagen 68.** Métrica del error cuadrático medio (MSE) para la colección de geometrías desconocidas, por función de pérdida.

Por su parte, en la imagen 69 se destaca nuevamente la función de pérdida  $\mathcal{L}_{\mathcal{F}_3}$ , quien alcanza el máximo global en la métrica SSIM, pero también vale la pena resaltar a la función de pérdida  $\mathcal{L}_{\mathcal{F}_2}$ , quien presenta un buen desempeño a través de las diferentes iteraciones.



**Imagen 69.** Métrica del Índice de similitud estructural (SSIM) para la colección de geometrías desconocidas, por función de pérdida.

Finalmente, en la imagen 70 nuevamente se observa un buen desempeño general para las funciones de pérdida  $\mathcal{L}\mathcal{F}_2$  y  $\mathcal{L}\mathcal{F}_3$  dentro de la métrica PSNR.



**Imagen 70.** Métrica de la Proporción máxima de señal a ruido (PSNR) para la colección de geometrías desconocidas, por función de pérdida.

### 5.5.4 Selección

Partiendo de los resultados previamente mostrados, sería posible encontrar un predominio en la calidad de la predicción de imágenes desconocidas con las funciones de pérdida  $\mathcal{L}\mathcal{F}_2$  y  $\mathcal{L}\mathcal{F}_3$ , quienes presentan los mejores valores dentro de las métricas de calidad de imagen consideradas.

La función de pérdida  $\mathcal{L}\mathcal{F}_2$  muestra un mínimo local dentro del  $MSE = 0.0043$  y máximos locales en las métricas  $SSIM = 0.757$  y  $PSNR = 18.32$ .

Por su parte, la función de pérdida  $\mathcal{L}\mathcal{F}_3$  muestra un mínimo local dentro del  $MSE = 0.0045$  y máximos globales en las métricas  $SSIM = 0.761$  y  $PSNR = 18.34$ .

Teniendo esto en cuenta, se debe realizar una selección entre estas arquitecturas para continuar con la optimización de las componentes de la red (sección 5.6).

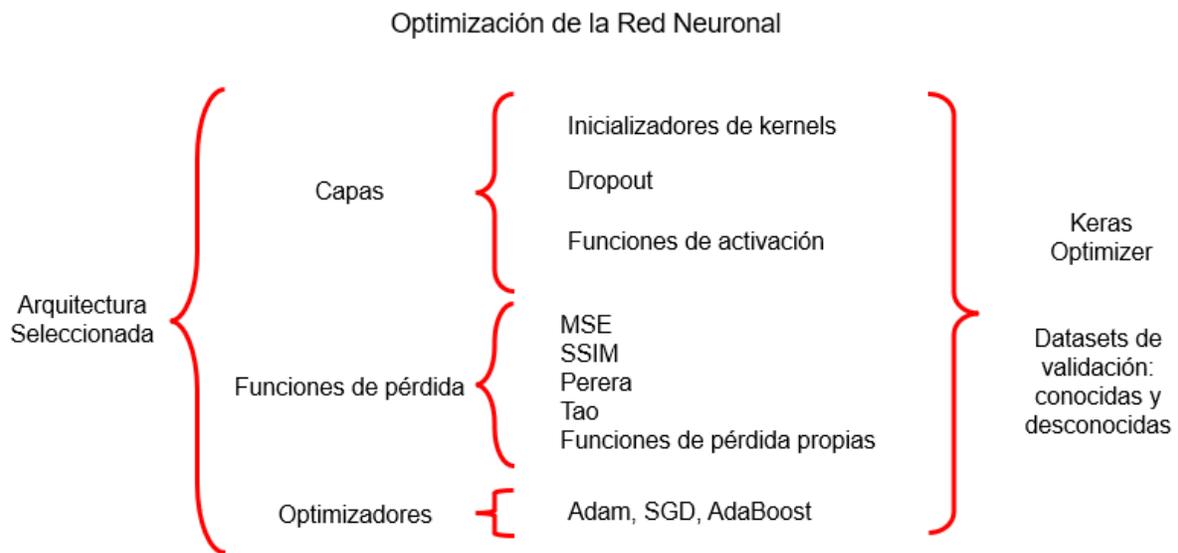
Siendo así, se determina que la función de pérdida a implementar es la función  $\mathcal{L}\mathcal{F}_3$ , debido a los resultados obtenidos en las métricas SSIM y PSNR.

## 5.6 Optimización de hiper-parámetros

Cuando ya se han seleccionado la arquitectura base y la función de pérdida, fundamentadas en los resultados mostrados, se continua con la optimización de las componentes del modelo de red neuronal.

Esta optimización de componentes se realiza por medio de experimentos en los que se configuran diferentes alternativas para una misma variable y se entrenan modelos de red neuronal, manteniendo las demás componentes con valores iniciales fijos.

La imagen 71 ilustra el proceso propuesto para la optimización de las componentes de la red:



**Imagen 71.** Variables de optimización del modelo de red neuronal.

La selección de la arquitectura y de la función de pérdida se realizó de manera manual e iterativa, en las secciones 5.4 y 5.5, respectivamente. Sin embargo, para la selección y optimización de las demás variables consideradas se utiliza la librería Keras Tuner.

A continuación, se comienza explicando brevemente el funcionamiento de Keras Tuner y se continua con el desarrollo de experimentos para la selección de demás hiper-parámetros. Inicialmente, se plantea una optimización paso a paso, en la que solo uno de los hiper-parámetros varía, mientras los demás permanecen con sus valores por defecto. Posteriormente, se plantea una optimización conjunta, en la que todos los hiper-parámetros considerados varían de manera simultánea, dentro de unos rangos establecidos.

### 5.6.1 Keras Tuner

Keras Tuner es una librería de código abierto que ofrece una solución eficiente para optimizar automáticamente los hiper-parámetros de modelos de aprendizaje profundo desarrollados con Keras y Tensorflow.

La optimización de hiper-parámetros es un proceso crítico en el desarrollo de modelos de aprendizaje automático, ya que afecta directamente el rendimiento y la generalización del modelo.

Al utilizar esta librería, se ahorra tiempo y esfuerzo, ya que la biblioteca automatiza gran parte del proceso de selección de hiper-parámetros, evitando así procesos manuales y permitiendo el enfoque en el diseño y la arquitectura del modelo en sí.

Los pasos para utilizar el optimizador de Keras son los siguientes:

- **Definir un espacio de búsqueda:** Este espacio incluye las diferentes opciones que se probarán para cada hiper-parámetro. Este espacio de búsqueda puede declararse en modo de lista finita o bien, un rango posible de variables numéricas.
- **Técnica de optimización:** Se pueden utilizar diversas técnicas de optimización para explorar el espacio de búsqueda. Algunas de las técnicas comunes incluyen la búsqueda aleatoria, la búsqueda en cuadrícula y la búsqueda de hiper-bandas. La búsqueda aleatoria elige combinaciones aleatorias de hiper-parámetros, mientras que la búsqueda en cuadrícula prueba todas las combinaciones posibles en un espacio discreto. Por su parte, la búsqueda de hiper-bandas es una técnica de optimización adaptativa que asigna recursos de computación de manera más eficiente al probar diferentes combinaciones de hiper-parámetros.
- **Evaluación del rendimiento:** Para cada combinación de hiper-parámetros, se entrena y evalúa un modelo en un conjunto de datos de validación. La métrica de rendimiento definida, como precisión o pérdida, se utiliza para evaluar qué combinación de hiper-parámetros produce el mejor modelo.
- **Búsqueda y refinamiento:** Keras Tuner itera a través del espacio de búsqueda durante un número predefinido de veces o hasta que se alcance un criterio de terminación. A medida que se evalúan diferentes combinaciones, la librería ajusta su enfoque para explorar más áreas prometedoras del espacio de hiper-parámetros.
- **Selección del mejor modelo:** Una vez que se completa la búsqueda, Keras Tuner devuelve el mejor modelo, junto con los hiper-parámetros que llevaron a su rendimiento óptimo.

Una vez se conoce el funcionamiento de este optimizador, se realizan entonces experimentos para encontrar los mejores valores de los distintos hiper-parámetros tenidos en consideración.

### 5.6.2 Experimento para el dropout

El dropout es una técnica de regularización utilizada para prevenir el sobreajuste (también llamado *overfitting*). El sobreajuste ocurre cuando el modelo se ajusta demasiado a los datos de entrenamiento y, como resultado, tiene un rendimiento deficiente en datos nuevos y no vistos.

El dropout es una forma de aleatorizar la red neuronal durante el entrenamiento, lo que ayuda a mejorar su capacidad para generalizar ante datos desconocidos. Consiste en "apagar" aleatoriamente un conjunto de neuronas durante cada paso de entrenamiento.

Estas neuronas apagadas no se utilizan para el cálculo de los gradientes y la actualización de los pesos, lo que las hace temporalmente inactivas.

En una red neuronal convolucional, las capas dropout se insertan después de las capas convolucionales. La capa dropout se define mediante un hiper-parámetro llamado "probabilidad de dropout". La probabilidad de dropout representa la fracción de neuronas que se desactivarán aleatoriamente durante el entrenamiento.

En el experimento planteado, se agregan capas de dropout luego de cada una de las capas convolucionales que conforman a la arquitectura VGG. Se utiliza la función de pérdida  $\mathcal{L}\mathcal{F}_3$  y se configuran las demás variables con valores por defecto, es decir, se utiliza Adam como optimizador, ReLU como función de activación y 'glorot-uniform' como inicializador de kernels.

Se configura el experimento de modo que los valores a iterar para la probabilidad de dropout están en un rango de 0.0 a 0.15 con pasos de 0.03, en un total de seis valores posibles. Para cada valor, Keras Tuner entrena un modelo durante diez épocas y calcula la función de pérdida  $\mathcal{L}\mathcal{F}_3$  en la colección de validación. Finalmente, escoge como mejor modelo a aquel que logra el menor valor en la función de pérdida. La tabla 2 resume los valores seleccionados para este experimento:

**Tabla 2.** Valores del experimento para la optimización del dropout.

Experimento Dropout			
Función de pérdida	$\mathcal{L}\mathcal{F}_3$	Valor típico	0.0
Optimizador	Adam	Valores a iterar	[0 - 0.15]
Activación	ReLU	Paso	0.03
Inicializador Kernel	glorot_uniform	Épocas / Step	10

Luego de realizar las diferentes iteraciones, Keras Tuner arroja que la probabilidad de dropout con la que se alcanza un menor valor de la función de pérdida dentro de la colección de validación es  $dropout - rate = 0.03$ .

### 5.6.3 Experimento para el optimizador

Los optimizadores son algoritmos que se utilizan para ajustar los pesos y los sesgos de las neuronas con el objetivo de minimizar la función de pérdida durante el proceso de entrenamiento.

El proceso de entrenamiento en una red neuronal implica el cálculo del gradiente de la función de pérdida con respecto a los parámetros del modelo (pesos y sesgos) para determinar cómo ajustar estos parámetros de manera óptima. Aquí es donde entran en juego los optimizadores, ya que se encargan de actualizar los parámetros de la red en función de los gradientes calculados.

Existen varios optimizadores populares utilizados en redes neuronales convolucionales, como el Descenso de Gradiente Estocástico (SGD), el método de Adagrad, el método de

RMSprop, el Algoritmo de Adam, entre otros. Cada optimizador tiene sus ventajas y desventajas, y el rendimiento puede variar según el problema y la arquitectura de la red.

En el experimento planteado, se agregan capas de dropout luego de cada una de las capas convolucionales que conforman a la arquitectura VGG, con una probabilidad de dropout de 0.03, teniendo en cuenta el resultado de la sección 5.6.2. Se utiliza la función de pérdida  $\mathcal{L}\mathcal{F}_3$  y se configuran las demás variables con valores por defecto, es decir, se utiliza ReLU como función de activación y 'glorot-uniform' como inicializador de kernels.

Se configura el experimento de modo que los valores a iterar para el optimizador están en una lista de seis posibles valores: Adam, SGD, RMSprop, Adadelata, Adagrad y Adamax. Para cada optimizador, Keras Tuner entrena un modelo durante diez épocas y calcula la función de pérdida  $\mathcal{L}\mathcal{F}_3$  en la colección de validación. Finalmente, escoge como mejor modelo a aquel que logra el menor valor en la función de pérdida. La tabla 3 resume los valores seleccionados para este experimento:

**Tabla 3.** Valores del experimento para la optimización del optimizador.

Experimento Optimizador			
Función de pérdida	$\mathcal{L}\mathcal{F}_3$	Valor típico	Adam
Dropout	0.03	Valores a iterar	Adam, SGD, RMSprop, Adadelata, Adagrad, Adamax.
Activación	ReLU	Paso	-
Inicializador Kernel	glorot_uniform	Épocas / Step	10

Luego de realizar las diferentes iteraciones, Keras Tuner arroja que el optimizador con el que se alcanza un menor valor de la función de pérdida dentro de la colección de validación es `optimizer = 'adam'`.

### 5.6.4 Experimento para las funciones de activación en las capas convolucionales

El funcionamiento de las funciones de activación en una red neuronal convolucional se basa en aplicar la función a la salida de cada neurona convolucional después de la operación de convolución. Esto introduce la no linealidad necesaria para que la red pueda aprender patrones complejos y relaciones no lineales en los datos. Cada función de activación tiene sus ventajas y desventajas. Dentro de las funciones de activación más comunes y que serán tenidas en cuenta en este experimento, se encuentran:

- **Función ReLU (Rectified Linear Unit):** Es una función simple y popular que devuelve 0 para todos los valores negativos de entrada y el mismo valor para valores positivos. Es la función de activación más utilizada en CNN debido a su eficiencia computacional y porque ayuda a resolver el problema del desvanecimiento del gradiente, permitiendo un entrenamiento más rápido y estable.

- **Función ELU** (*Exponential Linear Unit*): Similar a ReLU para valores positivos de entrada, es decir, es lineal. Sin embargo, para valores negativos de entrada, ELU se vuelve suave, lo que evita el problema de "neurona muerta" que puede ocurrir con ReLU. Además, la función ELU puede tomar valores negativos.
- **Función SELU** (*Scaled Exponential Linear Unit*): Es una versión escalada de ELU. Es una función de activación auto-normalizable, es decir, en una red profunda, las salidas de cada capa tienen una media y una varianza constante, lo que ayuda a mantener una propagación de señal estable y evita el problema del desvanecimiento o explosión del gradiente.
- **Función de activación Leaky ReLU**: Similar a la ReLU, pero en lugar de ser 0 para valores negativos, tiene una pendiente pequeña, lo que ayuda a resolver el problema del "neurona muerta" que puede surgir en ReLU cuando algunas neuronas nunca se activan.
- **Función de activación Sigmoide**: Transforma los valores de entrada en el rango de 0 a 1, lo que es útil para problemas de clasificación binaria. Sin embargo, puede sufrir del problema de desvanecimiento del gradiente en redes más profundas, lo que dificulta el entrenamiento efectivo de estas redes.
- **Función de activación Tanh** (tangente hiperbólica): Similar a la sigmoide, aplana los valores de entrada en un rango entre -1 y 1. Aunque evita algunos problemas de la sigmoide, aún puede experimentar el desvanecimiento del gradiente en redes más profundas.

En el experimento planteado, se agregan capas de dropout luego de cada una de las capas convolucionales que conforman a la arquitectura VGG, con una probabilidad de dropout de 0.03, teniendo en cuenta el resultado de la sección 5.6.2, se utiliza Adam como optimizador, teniendo en cuenta el resultado de la sección 5.6.3, se utiliza la función de pérdida  $\mathcal{L}_{\mathcal{F}_3}$  y se configuran las demás variables con valores por defecto, es decir, se utiliza 'glorot-uniform' como inicializador de kernels.

Para cada función de activación considerada, Keras Tuner entrena un modelo durante diez épocas y calcula la función de pérdida  $\mathcal{L}_{\mathcal{F}_3}$  en la colección de validación. Finalmente, escoge como mejor modelo a aquel que logra el menor valor en la función de pérdida. La tabla 4 resume los valores seleccionados para este experimento:

**Tabla 4.** Valores del experimento para la optimización de las funciones de activación.

Experimento Función de activación			
Función de pérdida	$\mathcal{L}_{\mathcal{F}_3}$	Valor típico	ReLU
Dropout	0.03	Valores a iterar	ReLU, ELU, SELU, Leaky ReLU, Sigmoide, Tanh
Optimizador	Adam	Paso	-
Inicializador Kernel	glorot_uniform	Épocas / Step	10

Luego de realizar las diferentes iteraciones, Keras Tuner arroja que la función de activación con la que se alcanza un menor valor de la función de pérdida dentro de la colección de validación es *activation = 'sigmoid'*.

### 5.6.5 Experimento para el inicializador de kernels

Los inicializadores de kernels son funciones que se utilizan para establecer los valores iniciales de los pesos en las capas convolucionales. Estos pesos son esenciales, ya que determinan cómo la red aprenderá a realizar la tarea específica para la que está siendo entrenada. Un buen conjunto inicial de pesos puede acelerar el proceso de entrenamiento y mejorar la convergencia, lo que lleva a un mejor rendimiento del modelo.

Su objetivo es evitar que los pesos se inicialicen con valores muy grandes o muy pequeños, lo que podría llevar a problemas de saturación (si son grandes) o a un rendimiento insatisfactorio (si son pequeños). También ayudan a introducir algo de aleatoriedad en la inicialización para evitar que todos los pesos comiencen con el mismo valor, lo que podría conducir a la simetría en la actualización de los pesos durante el entrenamiento. Algunos de los inicializadores más comunes y que serán tenidos en cuenta en este experimento, son:

- **Inicializador aleatorio normal** (*random\_normal*): Inicializa los pesos con valores aleatorios extraídos de una distribución normal (Gaussiana) con media cero y una desviación estándar determinada.
- **Inicializador de Glorot** (*glorot\_uniform*): Ajusta los pesos iniciales de acuerdo con el número de neuronas de entrada y salida de la capa. Ayuda a mantener una varianza constante a lo largo de las capas, mejorando el flujo de información hacia adelante y hacia atrás.
- **Inicializador de He** (*he\_normal*): Similar al inicializador de Glorot, pero ajusta los pesos iniciales de acuerdo con el número de neuronas de entrada solamente (no considera el número de neuronas de salida).

En el experimento planteado, se agregan capas de dropout luego de cada una de las capas convolucionales que conforman a la arquitectura VGG, con una probabilidad de dropout de 0.03, teniendo en cuenta el resultado de la sección 5.6.2, se utiliza Adam como optimizador, teniendo en cuenta el resultado de la sección 5.6.3, se utiliza la función sigmoide como función de activación de las capas convolucionales, teniendo en cuenta el resultado de la sección 5.6.4 y se utiliza la función de pérdida  $\mathcal{L}\mathcal{F}_3$ .

Para cada inicializador de kernels considerado, Keras Tuner entrena un modelo durante diez épocas y calcula la función de pérdida  $\mathcal{L}\mathcal{F}_3$  en la colección de validación. Finalmente, escoge como mejor modelo a aquel que logra el menor valor en la función de pérdida. La tabla 5 resume los valores seleccionados para este experimento:

**Tabla 5.** Valores del experimento para la optimización de los inicializadores de kernels.

Experimento Inicializadores de Kernels			
Función de pérdida	$\mathcal{L}\mathcal{F}_3$	Valor típico	glorot-uniform
Dropout	0.03	Valores a iterar	random_normal, he_normal, glorot_uniform
Optimizador	Adam	Paso	-
Activación	Sigmoid	Épocas / Step	10

Luego de realizar las diferentes iteraciones, Keras Tuner arroja que el inicializador de kernels con el que se alcanza un menor valor de la función de pérdida dentro de la colección de validación es  $kernel\_init = 'glorot\_uniform'$ .

### 5.6.6 Optimización conjunta

En las secciones anteriores (5.6.2 a 5.6.5) se realizó una optimización paso por paso, en la que cada uno de los hiper-parámetros se refinaba de manera independiente, manteniendo los demás hiper-parámetros con valores constantes.

En esta sección, se realiza un experimento en el que se iteran los cuatro hiper-parámetros considerados en conjunto, es decir, se permiten distintas combinaciones, manteniendo los mismos espacios de búsqueda que ya se utilizaron previamente. La tabla 6 resume las condiciones de este experimento:

**Tabla 6.** Valores del experimento para la optimización conjunta.

Experimento Conjunto				
Hiper-parámetro	Dropout	Optimizador	Activación	Kernel_init
Valor típico	0.0	Adam	ReLU	glorot_uniform
Valores a iterar	[0 - 0.15]	Adam, SGD, RMSprop, Adadelata, Adagrad, Adamax	ReLU, ELU, SELU, Leaky ReLU, Sigmoid, Tanh	random_normal, he_normal, glorot_uniform
Paso	0.03	-	-	-

Este experimento se corre por 20 iteraciones, en la que se entrena un modelo con diferentes combinaciones de hiper-parámetros por 10 épocas en cada iteración y se va almacenando la combinación que alcanza el menor valor en la función de pérdida dentro de la colección de validación.

El resultado arrojado por Keras Tuner es el siguiente (Tabla 7):

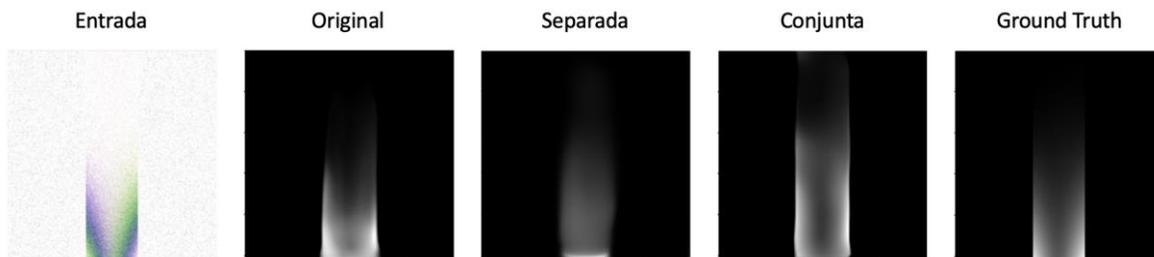
**Tabla 7.** Resultados de Keras Tuner para la optimización conjunta.

Hiper-parámetro	Valor
dropout_rate	0
activation	ReLU
Optimizer	Adam
Kernel_init	he_normal

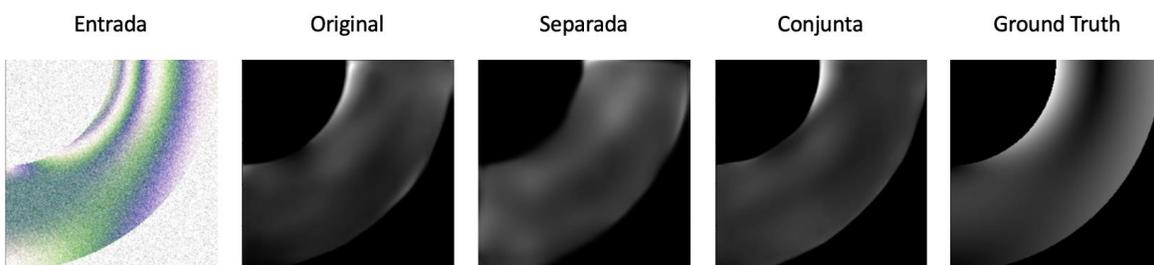
### 5.6.7 Entrenamiento de modelos y resultados

A partir de las configuraciones encontradas luego de realizar los procesos de optimización de hiper-parámetros, se entrenan dos posibles modelos finales: el primero con los resultados de la optimización paso a paso y el segundo, con los resultados de la optimización conjunta.

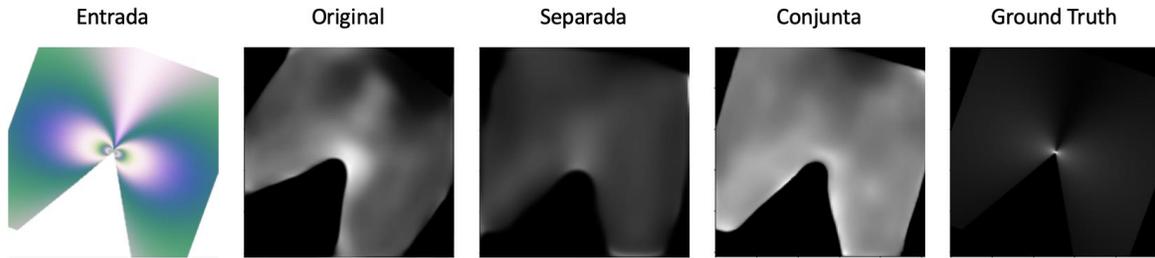
Ambos modelos se comparan a continuación con el modelo original, en el que no se había aplicado ningún proceso de optimización de hiper-parámetros. Las imágenes 72 a 74 ilustran los comparativos entre las predicciones de cada uno de estos modelos en las geometrías desconocidas:



**Imagen 72.** Predicciones sobre geometría de estructura vertical en modelos optimizados con Keras.



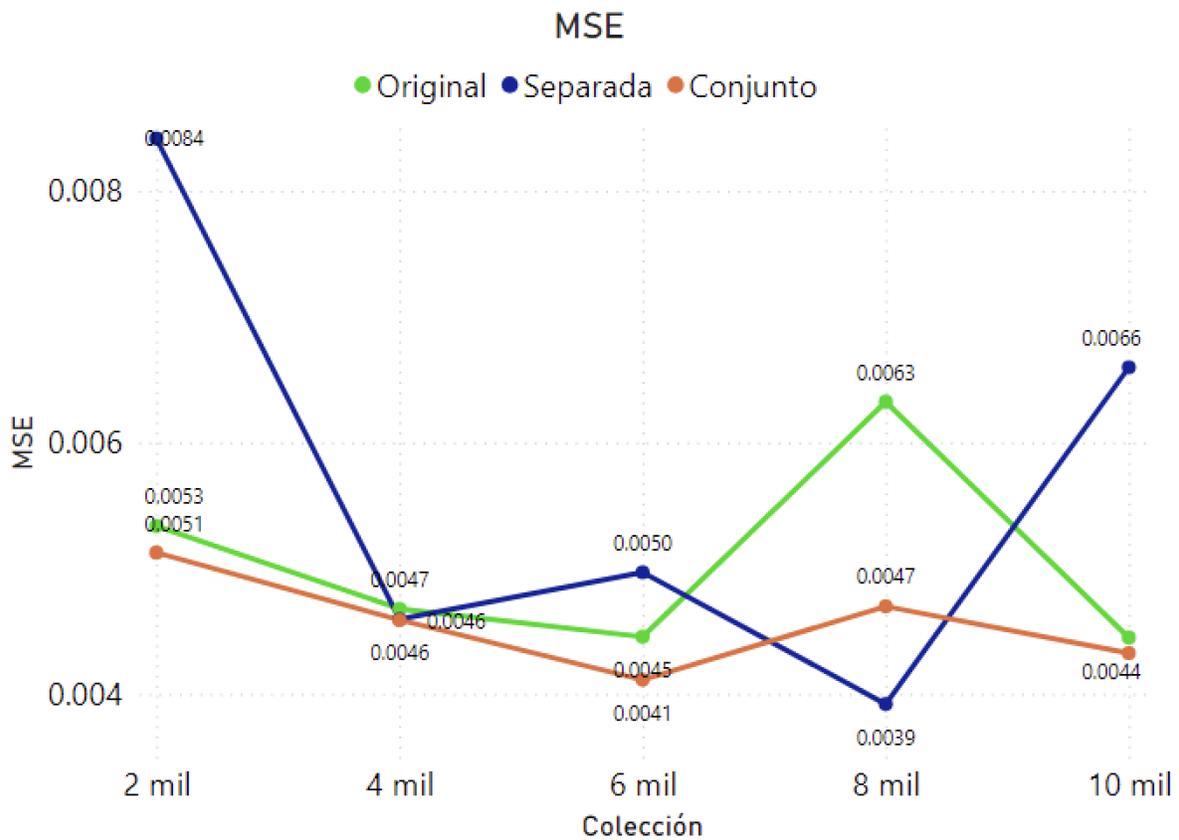
**Imagen 73.** Predicciones sobre geometría de ménsula en modelos optimizados con Keras.



**Imagen 74.** Predicciones sobre geometría de placa con muesca en V en modelos optimizados con Keras.

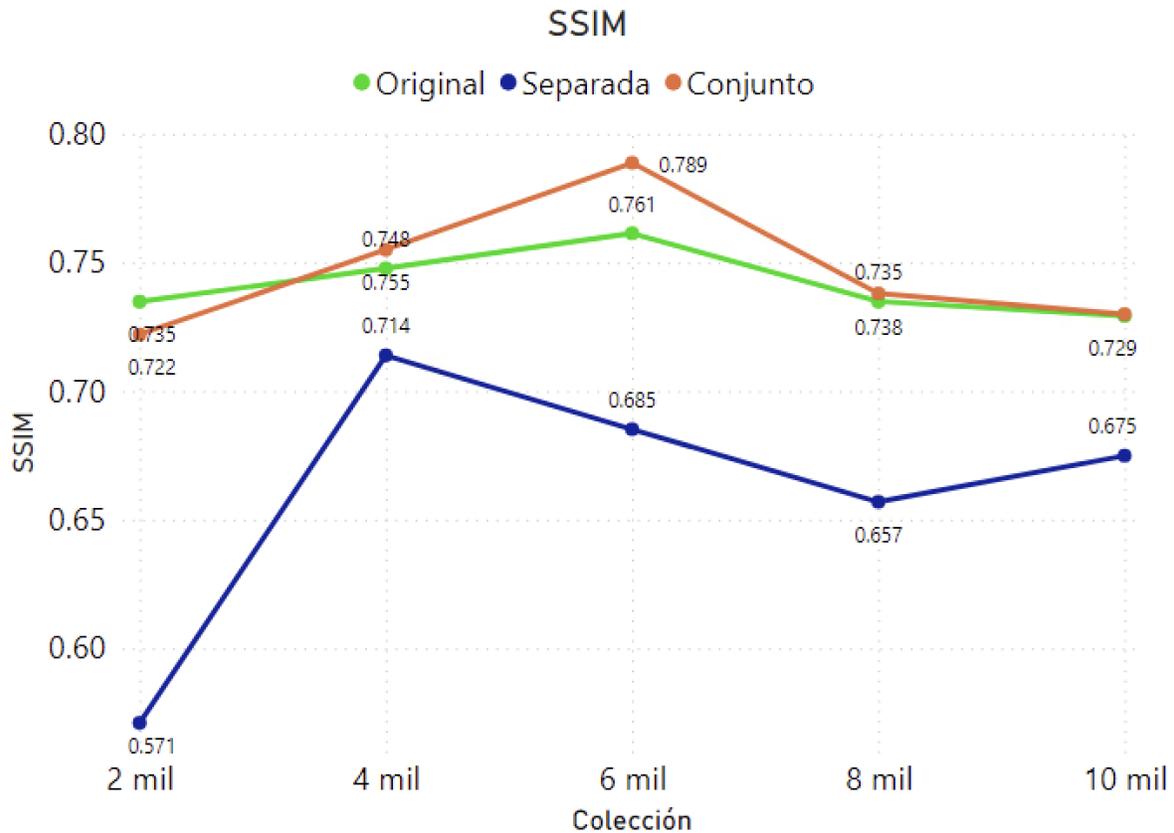
Las imágenes 75 a 77 ilustran las métricas alcanzadas en cada una de las iteraciones de validación.

En la imagen 75 se puede observar que el modelo optimizado en conjunto se caracteriza por un MSE relativamente estable y menor al del modelo original sin optimizar, mientras que el modelo optimizado por separado presenta un comportamiento más inestable que en algunas iteraciones se ubica por encima del MSE del modelo sin optimizar y en algunas otras por debajo.



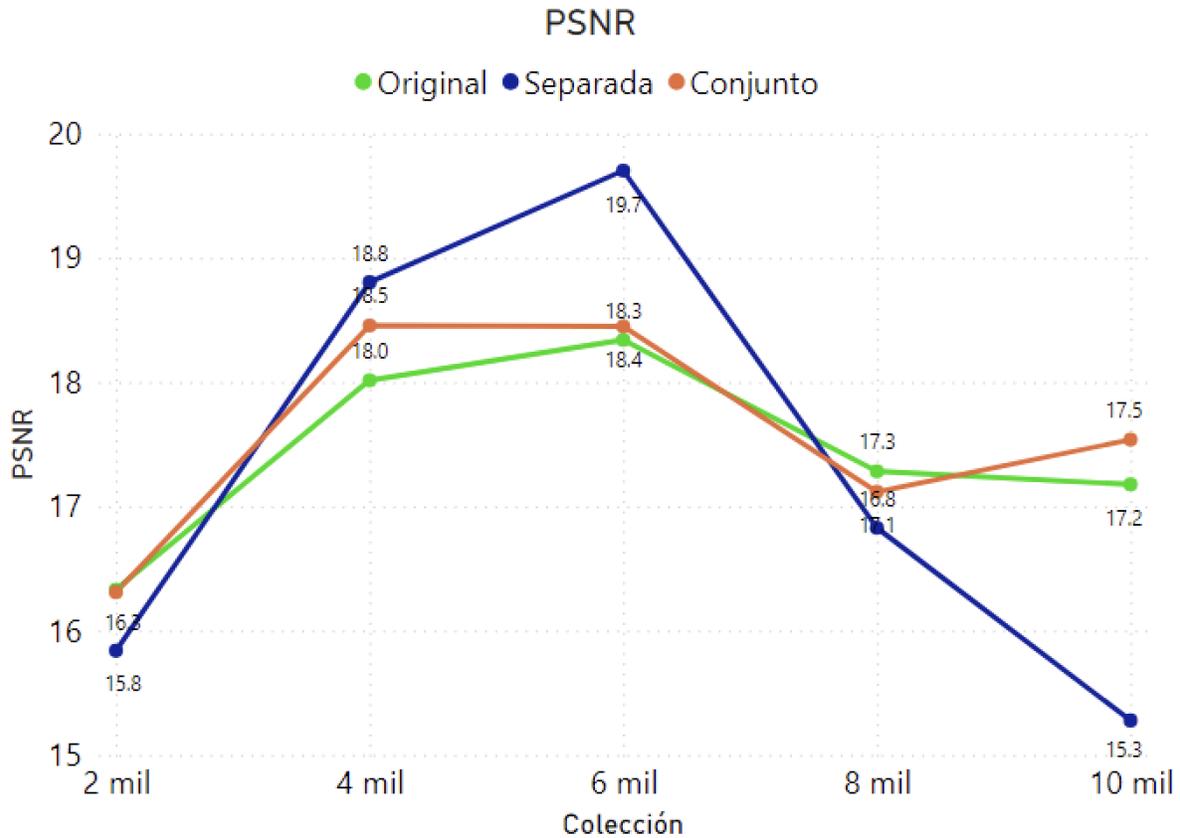
**Imagen 75.** Métrica del error cuadrático medio (MSE) para la colección de geometrías desconocidas, por modelo optimizado.

Por su parte, en la imagen 76 se evidencia un mejor desempeño en la métrica SSIM para el modelo optimizado en conjunto, alcanzando valores en su mayoría superiores a los del modelo sin optimizar. El modelo optimizado por separado alcanza valores inferiores en todas las iteraciones.



**Imagen 76.** Métrica del Índice de similitud estructural (SSIM) para la colección de geometrías desconocidas, por modelo optimizado.

Finalmente, en la imagen 77, se observa un desempeño bastante equivalente en los modelos original (sin optimizar) y modelo optimizado de manera conjunta. La optimización por separado alcanza valores pico, tanto máximos como mínimos.



**Imagen 77.** Métrica de la Proporción máxima de señal a ruido (PSNR) para la colección de geometrías desconocidas, por modelo optimizado.

### 5.6.8 Selección

De acuerdo con los resultados mostrados en la sección anterior, se hace evidente un predominio del modelo refinado mediante la optimización conjunta sobre el modelo refinado paso a paso, de manera separada.

Tanto en la métrica MSE como en la SSIM el modelo de la optimización conjunta logra alcanzar valores con mejor desempeño que la arquitectura sin optimizar.

Siendo así, se determina que el modelo refinado final, conocido como *MultipolarNet*, es el modelo que fue refinado de manera conjunta. La sección 5.7 profundiza esta red, sus características, componentes, resultados y métricas de calidad de imagen.

## 5.7 MultipolarNet

*MultipolarNet* es el nombre que se le ha dado a la red neuronal resultante de todo este proceso de investigación y desarrollo para la extracción de campos de esfuerzos en imágenes multipolarizadas de fotoelasticidad.

Como se mostró en las secciones anteriores, la construcción de esta red abarcó desde la construcción de colecciones de imágenes multipolarizadas para el entrenamiento y validación, pasando por una amplia investigación de diferentes arquitecturas clásicas que involucraban procesos de traducción de imágenes, desenvolvimiento de fase, procesamiento de imágenes con patrones de franjas, entre otras.

Además de ello, se realizó toda una experimentación para llevar a cabo una selección de la arquitectura base, de la función de pérdida y del proceso de optimización de hiper-parámetros. En cada uno de los experimentos se calcularon métricas de calidad de imagen y se obtuvieron predicciones con las cuales comparar de manera visual.

Esta sección resume las características de esta red, las componentes de su arquitectura, muestra algunas predicciones ante diferentes condiciones y finalmente expone sus métricas de calidad de imagen.

### 5.7.1 Características

Dentro de las características generales que describen a la MultipolarNet, se encuentran su cantidad de capas, las funciones de pérdida y de activación, algunos hiper-parámetros como la probabilidad de dropout, el optimizador utilizado, los inicializadores de kernels, entre otros. La tabla 8 resume las características más relevantes de la MultipolarNet.

**Tabla 8.** Características de la MultipolarNet.

Características MultipolarNet			
Capas Conv2D 3x3	10	Capas Encoder	5
Capas Conv2D 1x1	5	Capas Decoder	5
Dropout	0.0	Función de activación	ReLU
Strides	1	Optimizador	Adam
Padding	same	Inicializador Kernel	he_normal
Función de pérdida	$\mathcal{L}\mathcal{F}_3$	Parámetros No entrenables	9.858
Parámetros entrenables	45.341.187	Total parámetros	45.351.045

### 5.7.2 Arquitectura

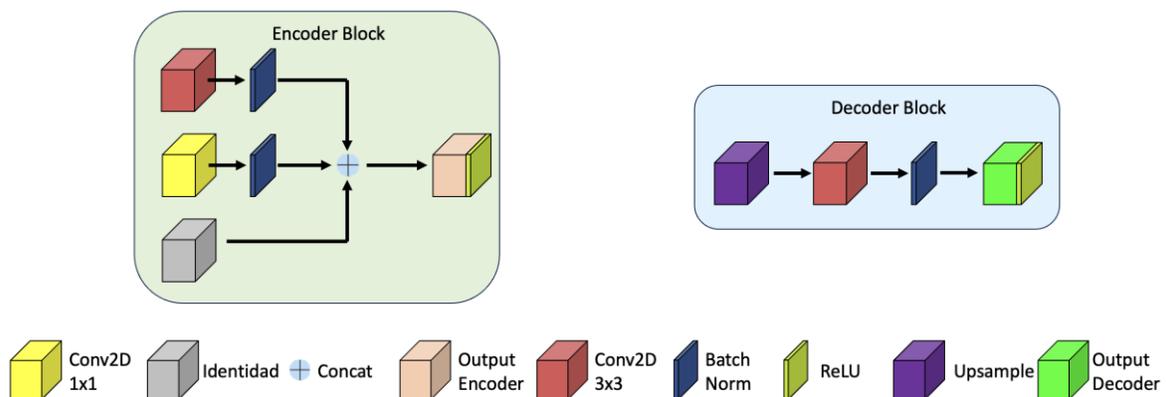
Como muestra previa de las capas y componentes de la MultipolarNet, la tabla 9 muestra la estructura general de la arquitectura. La capa de entrada puede adaptarse para recibir imágenes de tamaño (256, 256,3) en el caso de la representación Stokes o (256,256,4) en el caso de la representación Stack.

**Tabla 9.** Arquitectura MultipolarNet.

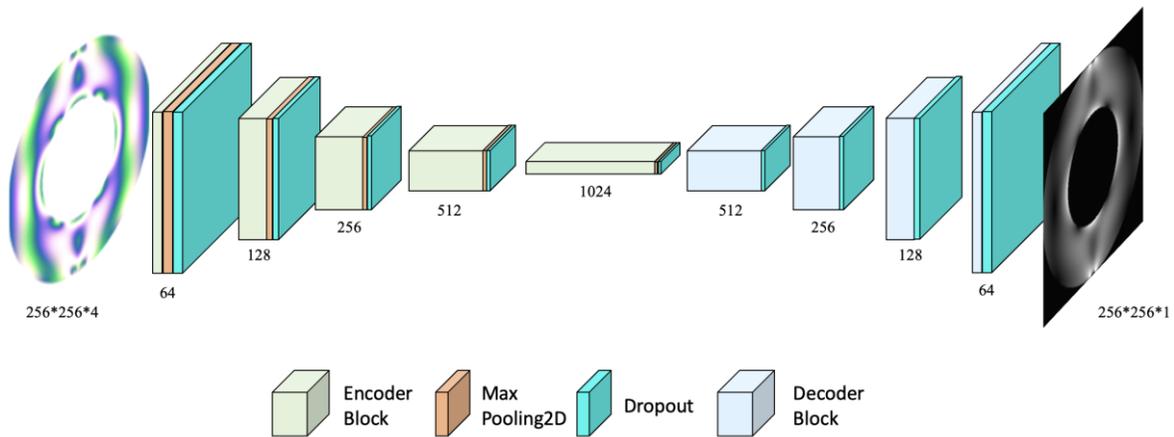
Capa	Filtros	Salida
Input		256, 256, 3 o 256, 256, 4
Conv2D 1,2	64	256, 256, 64
Batch Normalization 1,2		256, 256, 64
Concatenate 1		256, 256, 132
ReLU 1		256, 256, 132
MaxPooling2D 1		128, 128, 132
Dropout 1		128, 128, 132
Conv2D 3,4	128	128, 128, 128
Batch Normalization 3,4		128, 128, 128
Concatenate 2		128, 128, 388
ReLU 2		128, 128, 388
MaxPooling2D 2		64, 64, 388
Dropout 2		64, 64, 388
Conv2D 5,6	256	64, 64, 256
Batch Normalization 5,6		64, 64, 256
Concatenate 3		64, 64, 900
ReLU 3		64, 64, 900
MaxPooling2D 3		32, 32, 900
Dropout 3		32, 32, 900
Conv2D 7,8	512	32, 32, 512
Batch Normalization 7,8		32, 32, 512
Concatenate 4		32, 32, 1924
ReLU 4		32, 32, 1924
MaxPooling2D 4		16, 16, 1924
Dropout 4		16, 16, 1924
Conv2D 9,10	1024	16, 16, 1024
Batch Normalization 9,10		16, 16, 1024
Concatenate 5		16, 16, 3972
ReLU 5		16, 16, 3972
MaxPooling2D 5		8, 8, 3972
Dropout 5		8, 8, 3972
UpSampling 1		16, 16, 3972
Conv2D 11	512	16, 16, 512
Batch Normalization 11		16, 16, 512
ReLU 6		16, 16, 512
Dropout 6		16, 16, 512

UpSampling 2		32, 32, 512
Conv2D 12	256	32, 32, 256
Batch Normalization 12		32, 32, 256
ReLU 7		32, 32, 256
Dropout 7		32, 32, 256
UpSampling 3		64, 64, 256
Conv2D 13	128	64, 64, 128
Batch Normalization 13		64, 64, 128
ReLU 8		64, 64, 128
Dropout 8		64, 64, 128
UpSampling 4		128, 128, 128
Conv2D 14	64	128, 128, 64
Batch Normalization 14		128, 128, 64
ReLU 9		128, 128, 64
Dropout 9		128, 128, 64
UpSampling 5		256, 256, 64
Conv2D 15	1	256, 256, 1
Batch Normalization 15		256, 256, 1
ReLU 10		256, 256, 1
Output		256, 256, 1
<b>Total parámetros</b>		<b>45.351.045</b>
<b>Parámetros entrenables</b>		<b>45.341.187</b>
<b>Parámetros no entrenables</b>		<b>9.858</b>

Una representación gráfica de la MultipolarNet se muestra a continuación, en las imágenes 78 y 79:



**Imagen 78.** Representación bloques encoder y decoder de la MultipolarNet.



**Imagen 79.** Representación de la arquitectura de la MultipolarNet a partir de los bloques encoder y decoder definidos en la imagen 78.

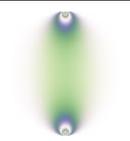
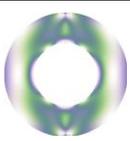
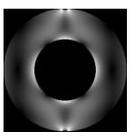
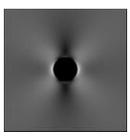
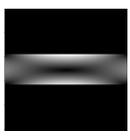
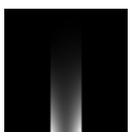
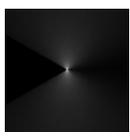
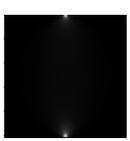
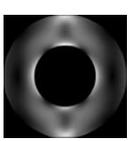
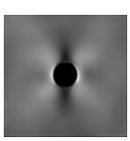
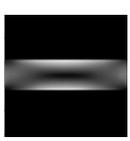
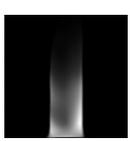
### 5.7.3 Predicciones

En esta sección se ilustran algunos de los resultados obtenidos por la MultipolarNet ante diferentes condiciones de carga (alta, media y baja), ante la presencia de alteraciones (ruido, brillo y contraste, rotaciones y recortes) y finalmente se realiza un comparativo de los resultados obtenidos al utilizar diferentes fuentes de iluminación o diferentes sensores de cámara polarizada. Las tablas 10, 11 y 12 muestran los resultados con diferentes cargas:

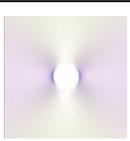
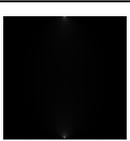
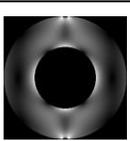
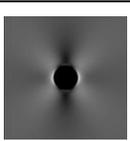
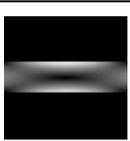
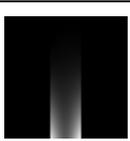
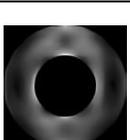
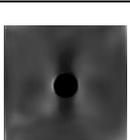
**Tabla 10.** Predicciones de la MultipolarNet ante cargas altas.

Predicciones ante cargas altas							
	Disco	Anillo	Placa	Viga	Estructura Vertical	Ménsula	Muesca
Imagen							
Objetivo							
Predicción							
MSE	9.862e-5	2.60e-3	1.384e-2	1.33e-3	3.22e-3	1.30e-2	9.22e-3
SSIM	0.988	0.940	0.901	0.984	0.906	0.772	0.773
PSNR	39.690	25.282	16.663	28.560	18.559	14.589	14.431

**Tabla 11.** Predicciones de la MultipolarNet ante cargas medias.

Predicciones ante cargas medias							
	Disco	Anillo	Placa	Viga	Estructura Vertical	Ménsula	Muesca
Imagen							
Objetivo							
Predicción							
MSE	7.872e-5	4.385e-4	2.822e-4	2.387e-4	1.113e-3	2.883e-3	1.060e-2
SSIM	0.993	0.958	0.972	0.990	0.916	0.809	0.697
PSNR	40.974	27.155	27.637	29.379	22.875	17.891	14.430

**Tabla 12.** Predicciones de la MultipolarNet ante cargas bajas.

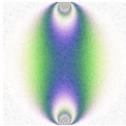
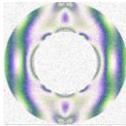
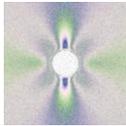
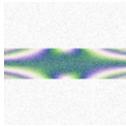
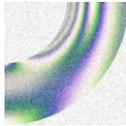
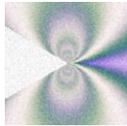
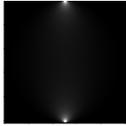
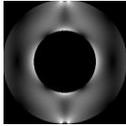
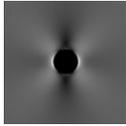
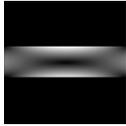
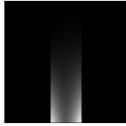
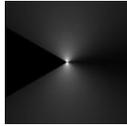
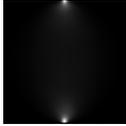
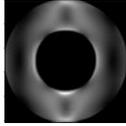
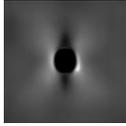
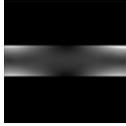
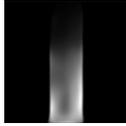
Predicciones ante cargas bajas							
	Disco	Anillo	Placa	Viga	Estructura Vertical	Ménsula	Muesca
Imagen							
Objetivo							
Predicción							
MSE	1.336e-5	2.956e-5	6.494e-5	1.455e-5	2.235e-4	2.032e-4	8.819e-4
SSIM	0.959	0.914	0.924	0.968	0.928	0.863	0.722
PSNR	36.816	24.568	23.988	28.557	24.153	21.870	15.060

De los resultados observados ante las diferentes cargas se puede evidenciar que la MultipolarNet es capaz de extraer campos en todas ellas, logrando buenos resultados en todas las métricas y geometrías, tanto conocidas como desconocidas.

En especial, podrían evidenciarse los mejores resultados en cargas promedio y cargas altas, a pesar de que también se obtienen buenas métricas en las cargas bajas, donde en algunas imágenes los patrones de franjas son bastante limitados.

A continuación, se muestran los resultados obtenidos por la MultipolarNet ante imágenes generadas en el proceso de aumento de datos. Las tablas 13, 14, 15 y 16 ilustran los resultados de las predicciones en imágenes con alteraciones por ruido, brillo, rotaciones y recortes, respectivamente:

**Tabla 13.** Predicciones de la MultipolarNet ante alteraciones de ruido.

Predicciones ante alteraciones de ruido							
	Disco	Anillo	Placa	Viga	Estructur a Vertical	Ménsula	Muesca
Imagen							
Objetivo							
Predicción							
MSE	7.569e-5	2.660e-3	1.763e-2	5.245e-4	3.219e-3	1.332e-2	7.268e-3
SSIM	0.993	0.938	0.876	0.990	0.909	0.761	0.782
PSNR	40.855	25.124	15.286	32.976	18.828	14.351	15.814

**Tabla 14.** Predicciones de la MultipolarNet ante alteraciones de brillo y contraste.

Predicciones ante alteraciones de brillo y contraste							
	Disco	Anillo	Placa	Viga	Estructura Vertical	Ménsula	Muesca
Imagen							
Objetivo							
Predicción							
MSE	7.143e-5	2.817e-3	2.052e-2	5.776e-4	2.914e-3	1.384e-2	9.585e-3
SSIM	0.995	0.938	0.857	0.989	0.895	0.756	0.772
PSNR	41.140	25.085	15.550	32.390	20.517	14.015	15.486

**Tabla 15.** Predicciones de la MultipolarNet ante alteraciones de rotaciones.

Predicciones ante alteraciones de rotaciones							
	Disco	Anillo	Placa	Viga	Estructura Vertical	Ménsula	Muesca
Imagen							
Objetivo							
Predicción							
MSE	1.207e-3	2.971e-3	8.999e-3	3.494e-3	3.233e-3	9.346e-3	8.961e-3
SSIM	0.972	0.927	0.906	0.943	0.868	0.787	0.775
PSNR	27.679	24.780	19.160	23.167	19.597	13.778	15.325

**Tabla 16.** Predicciones de la MultipolarNet ante alteraciones de recortes.

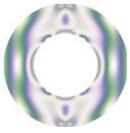
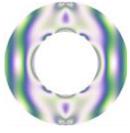
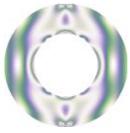
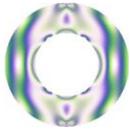
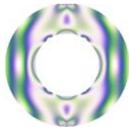
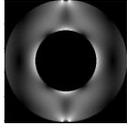
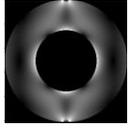
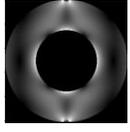
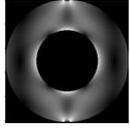
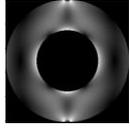
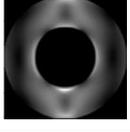
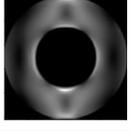
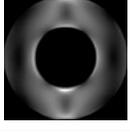
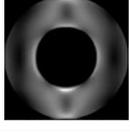
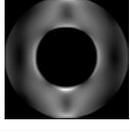
Predicciones ante alteraciones de recortes							
	Disco	Anillo	Placa	Viga	Estructura Vertical	Ménsula	Muesca
Imagen							
Objetivo							
Predicción							
MSE	1.105e-4	4.617e-3	2.719e-2	1.319e-3	1.064e-2	1.379e-2	8.577e-3
SSIM	0.984	0.954	0.809	0.987	0.902	0.774	0.730
PSNR	39.262	22.245	13.749	27.774	13.796	5.518	12.774

Estos resultados dan evidencia de la capacidad de la MultipolarNet de predecir campos de esfuerzo en imágenes multipolarizadas que han sufrido alteraciones, tanto en geometrías conocidas como desconocidas.

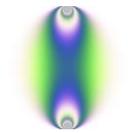
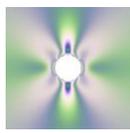
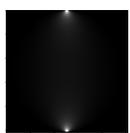
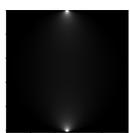
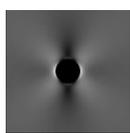
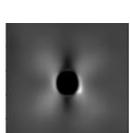
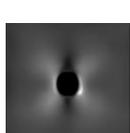
La MultipolarNet logra desenvolver los patrones de franjas con buenas métricas, a pesar de los diferentes procesos de transformación que se aplicaron en el aumento de datos.

Finalmente, las tablas 17 y 18 realizan un comparativo de los resultados obtenidos por la MultipolarNet ante imágenes multipolarizadas compuestas por diferentes fuentes de iluminación y sensores de cámara, respectivamente:

**Tabla 17.** Predicciones de la MultipolarNet con diferentes fuentes de iluminación.

Predicciones con diferentes fuentes de iluminación					
	Concord	LCD	Fluorescente	Toshiba	Willard
Imagen					
Objetivo					
Predicción					
MSE	2.605e-3	2.568e-3	2.675e-3	2.694e-3	2.724e-3
SSIM	0.940	0.940	0.938	0.937	0.936
PSNR	25.282	25.385	25.154	25.133	25.079

**Tabla 18.** Predicciones de la MultipolarNet con diferentes sensores de cámara.

Predicciones con diferentes sensores de cámara						
	Disco Concord MYR	Disco Concord MZR	Placa Sylvania MYR	Placa Sylvania MZR	Ménsula LCD MYR	Ménsula LCD MZR
Imagen						
Objetivo						
Predicción						
MSE	9.862e-5	9.834e-5	1.438e-2	1.488e-2	1.206e-2	1.232e-2
SSIM	0.988	0.987	0.898	0.895	0.778	0.777
PSNR	39.690	39.754	16.655	16.719	14.976	14.903

De estos resultados se observa inicialmente que las fuentes de iluminación y los sensores polarizados utilizados en la construcción de las imágenes multipolarizadas generan imágenes bastante similares, cuyas diferencias se hacen incluso imperceptibles para el ojo humano.

Sin embargo, sí existen diferencias a la hora de predecir las imágenes, lo cual es observable en las métricas utilizadas, que presentan valores diferentes pero bastante cercanos.

Dentro de las fuentes de iluminación analizadas, se observa que la fuente Concord y la fuente LCD obtienen el valor más alto en la métrica SSIM, más la fuente LCD alcanza un valor más alto en la métrica PSNR y un menor MSE, por lo que podría afirmarse que es la fuente con mejores resultados.

Por su parte, las métricas de la comparación entre el sensor policromático MYR y el monocromático MZR también son bastante cercanas. Se observa que para las diferentes geometrías, la métrica SSIM alcanza valores un poco más altos en el sensor MYR, pero en algunas de ellas también obtiene valores inferiores en la métrica PSNR o valores más elevados en el MSE.

En general podría afirmarse que las fuentes de iluminación y los sensores de cámara acá analizados presentan un desempeño bastante similar, donde no juegan un rol determinante en la calidad de las predicciones ejecutadas por la MultipolarNet.

#### 5.7.4 Métricas de calidad de imagen

Con el objetivo de cuantificar el desempeño general de la MultipolarNet a través de sus diferentes etapas de entrenamiento, se utilizan diferentes métricas de calidad de imagen encontradas en el estado del arte, además de las métricas tradicionales.

Para cada etapa de entrenamiento, se toman paquetes de 2000 imágenes con geometrías desconocidas y se cuantifican las diferentes métricas en cada una de las imágenes generadas por la red. Finalmente se obtiene el promedio de cada métrica en cada uno de los paquetes y su desviación estándar.

La tabla 19 contiene los valores obtenidos para las métricas tradicionales: MSE, SSIM y PSNR.

**Tabla 19.** Promedio y desviación estándar por iteración de la MultipolarNet en métricas de calidad de imagen tradicionales sobre geometrías desconocidas.

Métricas tradicionales						
Iteración	mse_mean	mse_std	ssim_mean	ssim_std	psnr_mean	psnr_std
2000	5,125e-03	4,666e-03	0,722	0,198	16,311	5,204
4000	4,589e-03	4,638e-03	0,755	0,207	18,456	5,195
6000	4,113e-03	4,581e-03	0,789	0,207	18,459	5,887
8000	4,697e-03	4,109e-03	0,738	0,201	17,117	5,574
10000	4,325e-03	4,094e-03	0,730	0,209	17,537	5,669

De esta tabla es posible evidenciar una tendencia al decremento en el MSE y una estabilización en las métricas SSIM y PSNR con un pico máximo en el desempeño en el ciclo de 4000 a 6000.

Por su parte, se cuantifica también el desempeño de la MultipolarNet por medio de métricas adicionales, implementadas en la librería pyiqa, expuesta en la sección 5.2.3.

La tabla 20 contiene los valores obtenidos por la MultipolarNet en las métricas Full Reference FR, en las que se comparan las predicciones de la red con el Ground Truth. En este caso, tanto la métrica DISTS como la métrica LPIPS optimizan su valor hacia el mínimo.

**Tabla 20.** Promedio y desviación estándar por iteración de la MultipolarNet en métricas de calidad de imagen “Full Reference” (FR) sobre geometrías desconocidas.

Métricas Full Reference				
Iteración	dists_mean	dists_std	lpips_mean	lpips_std
2000	0,269	0,060	0,192	0,106
4000	0,258	0,067	0,177	0,106
6000	0,259	0,075	0,180	0,111
8000	0,256	0,067	0,169	0,104
10000	0,242	0,073	0,165	0,102

De esta tabla es posible evidenciar nuevamente una tendencia a la optimización a través de los ciclos de entrenamiento, pues tanto la métrica DISTS como la métrica LPIPS van disminuyendo su valor, alcanzando sus mínimos en la última iteración de entrenamiento.

Finalmente, las tablas 21 y 22 contienen los valores obtenidos por la MultipolarNet en las métricas No Reference NR, en las que diferentes modelos de Machine y de Deep Learning brindan una calificación del aspecto de la imagen obtenida por la red, basados en sus características y comparándolas con percepciones humanas.

Dentro de las métricas NR se encuentra la métrica BRISQUE que se optimiza hacia el mínimo y las métricas DBCNN, NIMA y CKDN, que se optimizan hacia el máximo:

**Tabla 21.** Promedio y desviación estándar por iteración de la MultipolarNet en métricas de calidad de imagen “No Reference” (FR) sobre imágenes desconocidas. (Parte I)

Métricas No Reference				
Iteración	brisque_mean	brisque_std	dbcnn_mean	dbcnn_std
2000	102,224	4,796	36,124	1,727
4000	101,602	6,016	35,949	1,876
6000	109,206	3,750	35,436	1,808
8000	109,988	4,599	36,398	2,047
10000	110,384	5,023	36,417	2,111

**Tabla 22.** Promedio y desviación estándar por iteración de la MultipolarNet en métricas de calidad de imagen “No Reference” (FR) sobre imágenes desconocidas. (Parte II)

Métricas No Reference				
Iteración	nima_mean	nima_std	ckdn_mean	ckdn_std
<b>2000</b>	4,392	0,923	0,504	0,028
<b>4000</b>	4,400	0,899	0,504	0,029
<b>6000</b>	4,374	0,821	0,501	0,029
<b>8000</b>	4,527	0,829	0,504	0,029
<b>10000</b>	4,541	0,897	0,506	0,027

De estas tablas es posible comprobar que la MultipolarNet se optimiza a través de las iteraciones de entrenamiento, con las métricas No Reference, pues se observa como las métricas DBCNN, NIMA y CKDN aumentan su valor a través de las iteraciones, alcanzando sus valores máximos en la última iteración realizada.

## 6. Conclusiones y trabajo futuro

Este trabajo desarrolla un modelo de red neuronal profunda para la extracción de campos de esfuerzos partiendo de imágenes multipolarizadas de fotoelasticidad.

A continuación, se enuncian las conclusiones y el trabajo futuro que se propone:

- Respecto al objetivo general, se desarrolló un modelo de red neuronal profunda para la extracción de campos de esfuerzos partiendo de imágenes de multipolarizadas de fotoelasticidad.
- Respecto a los objetivos específicos, se implementó una estrategia de generación de imágenes digitales que representa diferentes condiciones reales de los modelos de fotoelasticidad.
- Se estableció un método para determinar los hiper-parámetros de las arquitecturas de red neuronal, mediante experimentos manuales y librerías de optimización de modelos, como lo es Keras Tuner.
- Se evaluó el desempeño de las diferentes arquitecturas de red neuronal utilizando métricas de calidad de imagen y con ello se realizaron procesos de selección de arquitectura, función de pérdida y optimización de hiper-parámetros.
- Con respecto a las preguntas de investigación de esta tesis, sí es posible extraer campos de esfuerzos partiendo de imágenes multipolarizadas de fotoelasticidad digital y mediante el uso de redes neuronales profundas.
- En el artículo Photoelastic and Stokes Images through Deep Convolutional Neural Networks: A Comparison of Stress Fields, generado como complemento a esta tesis, se demostró que la representación apilada (Stack) obtiene mejores métricas de calidad de imagen que la representación Stokes.
- Según el experimento realizado para la selección de una arquitectura base y teniendo en cuenta algunas arquitecturas tradicionales como UNet, UNet con capas LSTM, StressNet, PhotoelastNet, VGG y GAN + Pix2Pix, la arquitectura que mejores métricas de calidad de imagen alcanzó fue la VGG, la cual fue seleccionada para continuar con su refinamiento.
- La arquitectura GAN + Pix2Pix presentó los resultados más bajos en cuanto a las métricas de calidad de imagen, no porque su arquitectura no fuese adecuada, sino porque quizá hacía falta un poco más de iteraciones durante el entrenamiento, ya que a pesar de ser el resultado que más se aleja a la predicción esperada en todas las geometrías, es posible observar que las predicciones que genera tienen la forma inicial esperada, por lo que, quizá con un mayor entrenamiento y refinamiento, los resultados podrían mejorar notablemente.
- Según el experimento realizado para la selección de una función de pérdida, utilizando la arquitectura VGG como base, se demostró que la función de pérdida ' $\mathcal{L}_{\mathcal{F}_3}$ ', compuesta a partir de los errores MSE, MAE y las métricas MAD, BPE y SSIM alcanzaba

el mejor desempeño en la predicción de campos de esfuerzo sobre geometrías desconocidas por la red.

- En general, las funciones de pérdida compuestas a partir de diferentes métricas de calidad de imagen y errores demostraron un mejor desempeño que las funciones de pérdida donde se utiliza una única métrica.
- Al realizar una optimización paso a paso por medio de Keras Tuner, en la que solo un hiper-parámetro se variaba mientras los demás permanecían constantes, se demostró que los valores óptimos eran: Dropout de 0.03, optimizador Adam, función de activación Sigmoide y glorot\_uniform como inicializador de kernels.
- Al realizar una optimización conjunta por medio de Keras Tuner, en la que todos los hiper-parámetros considerados variaban dentro de rangos especificados, se demostró que los valores óptimos eran: Dropout de 0.0, optimizador Adam, función de activación ReLU y he\_normal como inicializador de kernels.
- Al realizar una comparación entre los desempeños de ambas configuraciones propuestas por Keras Tuner (paso a paso y conjunta), se demostró que la configuración planteada por la optimización conjunta alcanzaba un mayor desempeño en la predicción de campos de esfuerzo.
- A partir de estos experimentos, se construyó y refinó la denominada *MultipolarNet*, la arquitectura de red neuronal con un desempeño más adecuado para la extracción de campos de esfuerzo en imágenes multipolarizadas de fotoelasticidad.
- La *MultipolarNet* es un modelo de red neuronal capaz de extraer campos de esfuerzos en imágenes multipolarizadas de fotoelasticidad con geometrías clásicas con las que fue entrenada (Disco, Anillo, Placa, Viga) y con geometrías desconocidas (Estructura vertical bajo presión, ménsula cargada en su extremo libre, placa con muesca en V).
- La *MultipolarNet* es capaz de extraer campos de esfuerzos en imágenes multipolarizadas de fotoelasticidad con geometrías bajo diferentes intensidades de carga (Baja, media y alta).
- La *MultipolarNet* es capaz de extraer campos de esfuerzos en imágenes multipolarizadas de fotoelasticidad con alteraciones tales como presencia de ruido, aumentos y variaciones en el brillo y el contraste, rotaciones y recortes de las imágenes.
- La *MultipolarNet* es capaz de extraer campos de esfuerzos en imágenes multipolarizadas que presenten isoclínicos y es capaz de generalizar ante imágenes de fotoelasticidad bajo diferentes condiciones de iluminación y diferentes sensores de cámara.
- La *MultipolarNet* presenta un comportamiento correcto en métricas de calidad de imagen tradicionales, Full Reference y No Reference, a través de las diferentes iteraciones de entrenamiento.

Como trabajo futuro en la ampliación de esta tesis, se puede partir del hecho de que las redes evaluadas en este trabajo representan una pequeña muestra del número de arquitecturas diferentes que podrían utilizarse para extraer campos de esfuerzo en

imágenes multipolarizadas, quedando aún una amplia gama de arquitecturas e hiper-parámetros por explorar y optimizar. La construcción de nuevas arquitecturas de red neuronal es un tema recurrente en la investigación actual, por lo que es probable que nuevas tendencias alcancen incluso mejores resultados.

Así mismo, queda una oportunidad latente en la generación de más geometrías analíticas de fotoelasticidad digital, así como del uso de las redes implementadas con imágenes multipolarizadas reales, que permitan la construcción de modelos más robustos y versátiles.

De igual manera, en este trabajo se utilizan dos representaciones de imágenes multipolarizadas: la representación apilada *4-stack* y *Stokes*, pero es importante resaltar la presencia en la literatura de algunas otras representaciones que podrían ser consideradas y comparadas con las presentadas en esta tesis.

Finalmente, sucede algo similar con las métricas de calidad de imagen e hiper-parámetros, que están en constante crecimiento y son factibles de ser regeneradas y optimizadas por la comunidad científica, aportando así a la construcción de modelos más precisos.

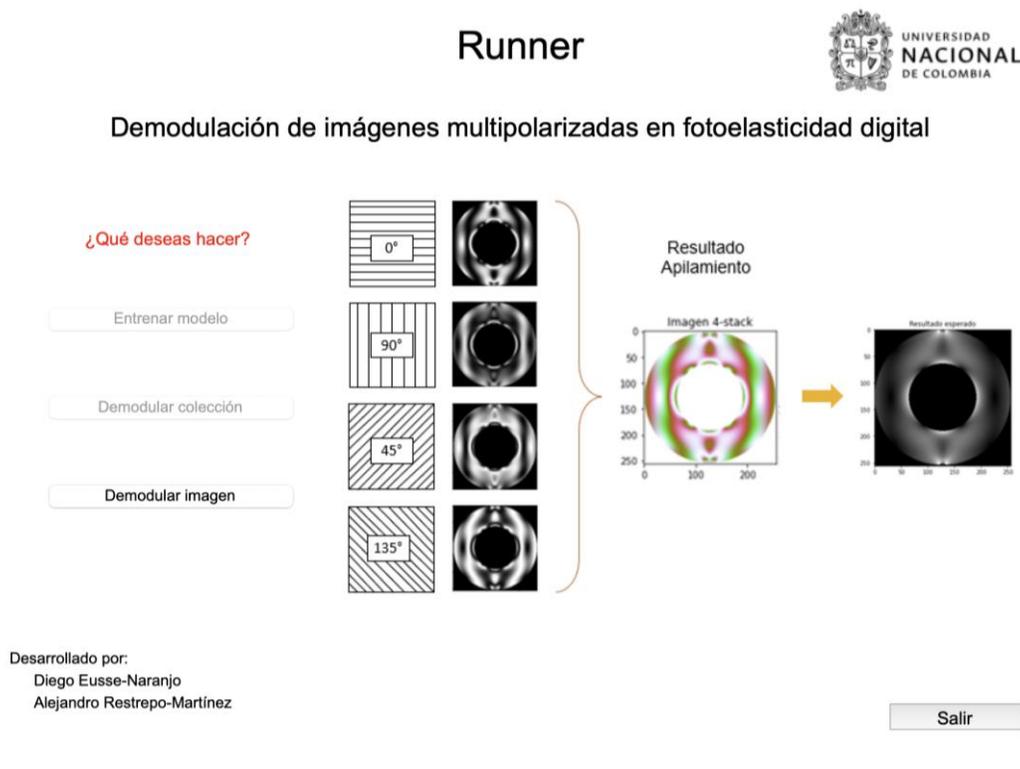
## 7. Anexos

En esta sección se presentan algunos códigos e implementaciones que hacen parte del desarrollo de toda esta tesis y que pueden ser útiles en caso de querer profundizar, continuar o corroborar los estudios que aquí se presentaron.

### 7.1 Anexo 1: Interfaz gráfica para el entrenamiento de modelos y evaluación por métricas de calidad de imagen

Como alternativa para realizar el entrenamiento y evaluación de arquitecturas de manera local y a través de un proceso parametrizado, se comenzó con el desarrollo de una interfaz gráfica sencilla que permita desde entrenar modelos, demodular colecciones completas de imágenes multipolarizadas, o bien, demodular una única imagen. Se planea también incluir el cálculo de las métricas de calidad de imagen de estas predicciones.

La imagen 80 ilustra la pantalla inicial de esta interfaz, que se encuentra en desarrollo:



**Imagen 80.** Interfaz gráfica para entrenamiento de redes y demodulación de imágenes multipolarizadas.

El código fuente de esta interfaz está siendo desarrollado en el siguiente repositorio: <https://github.com/diegoeusse/PhaseDemodulator>

## **7.2 Anexo 2: Código para la generación de la colección de imágenes multipolarizadas**

De manera similar, se crea un repositorio con los códigos utilizados en la generación de la colección de imágenes multipolarizadas, desde los códigos en Matlab en los que se configuran las ecuaciones de cada uno de los modelos, pasando por la construcción de las imágenes isocromáticas en cada uno de los estados de polarización, para luego apilar estas imágenes mediante Jupyter Notebooks y crear así una gran colección de imágenes multipolarizadas con diferentes alteraciones.

Entre estos códigos también se encuentra la base para la generación de las geometrías desconocidas y los códigos fuente para generar las imágenes Ground Truth.

El repositorio puede encontrarse en:

<https://github.com/diegoeusse/CreacionImágenesMultipolarizadas>

## **7.3 Anexo 3: Código base de arquitecturas analizadas**

Finalmente, se crea también un repositorio con los códigos fuentes de las arquitecturas analizadas en esta tesis, su estructura, sus procedimientos de entrenamiento, la visualización de algunos resultados y la cuantificación de sus métricas.

Estos códigos se desarrollan en su totalidad mediante Jupyter Notebooks, su organización es bastante similar y lo que varía netamente es la arquitectura de cada red neuronal.

El repositorio puede encontrarse en:

<https://github.com/diegoeusse/RedesNeuronalesDemodulacionImágenes>

## 8. Referencias

- [1] Juan C. Briñez-de León, Alejandro Restrepo-Martínez y Francisco López Giraldo, Estudios de fotoelasticidad: desarrollos y aplicaciones, Revista Politécnica ISSN 1900-2351, Año 9, Número 16, Páginas 27-36, 2013.
- [2] Juan C. Briñez-de León, Mateo Rico-García, and Alejandro Restrepo-Martínez, "PhotoelastNet: a deep convolutional neural network for evaluating the stress field by using a single-color photoelasticity image," Appl. Opt. 61, D50-D62 (2022).
- [3] Briñez de León, Juan & Rico, Mateo & Branch, John & Restrepo-Martínez, Alejandro. (2020). StressNet: A deep convolutional neural network for recovering the stress field from isochromatic images. 23. 10.1117/12.2568609.
- [4] Tao Bo, Wang Yan, Qian Xinbo, Tong Xiliang, He Fuqiang, Yao Weiping, Chen Bin, Chen Baojia. Photoelastic Stress Field Recovery Using Deep Convolutional Neural Network, Frontiers in Bioengineering and Biotechnology, Volume 10, 2022
- [5] Ren, Zhangyu & Xie, Huimin & Ju, Yang. (2020). Quantification of photoelastic fringe orders using polarized light camera and continuous loading. Optics and Lasers in Engineering. 134. 106263. 10.1016/j.optlaseng.2020.106263.
- [6] Dubey, V. & Grewal, Gurtej & Claremont, D. (2007). Load Extraction from Photoelastic Images Using Neural Networks. Experimental Mechanics. 47. 263-270. 10.1007/s11340-006-9002-z.
- [7] Ketao Yan, Yingjie Yu, Chongtian Huang, Liansheng Sui, Kemao Qian, Anand Asundi, Fringe pattern denoising based on deep learning, Optics Communications, Volume 437, 2019, Pages 148-152, ISSN 0030-4018.
- [8] Wang, Kaiqiang & Li, Ying & Kemao, Qian & Di, Jianglei & Zhao, Jianlin. (2019). One-step robust deep learning phase unwrapping. Optics Express. 27. 15100. 10.1364/OE.27.015100.
- [9] Teng Zhang, Shaowei Jiang, Zixin Zhao, Krishna Dixit, Xiaofei Zhou, Jia Hou, Yongbing Zhang, and Chenggang Yan, "Rapid and robust two-dimensional phase unwrapping via deep learning," Opt. Express 27, 23173-23185 (2019).
- [10] Junchao Zhang, Xiaobo Tian, Jianbo Shao, Haibo Luo, and Rongguang Liang, "Phase unwrapping in optical metrology via denoised and convolutional segmentation networks," Opt. Express 27, 14903-14912 (2019).
- [11] Lin, Bowen & Fu, Shujun & Zhang, Caiming & Wang, Fengling & Li, Yuliang. (2019). Optical Fringe Patterns Filtering Based on Multi-Stage Convolution Neural Network.
- [12] Juan C. Briñez de León, Ruben D. Fonnegra-Tarazona, and Alejandro Restrepo-Martínez "Generalized adversarial networks for stress field recovering processes from photoelasticity images", Proc. SPIE 11510, Applications of Digital Image Processing XLIII, 115100S (21 August 2020).

**[13]** Ketao Yan, Yingjie Yu, Tao Sun, Anand Asundi, Qian Kemao, Wrapped phase denoising using convolutional neural networks, *Optics and Lasers in Engineering*, Volume 128, 2020.

**[14]** Perera, Malsha & de Silva, Ashwin. (2020). A Joint Convolutional and Spatial Quad-Directional LSTM Network for Phase Unwrapping. *IEEE International Conference on Acoustics, Speech and Signal Processing. (ICASSP)*.

**[15]** Alan Reyes-Figueroa, Victor H. Flores, and Mariano Rivera, "Deep neural network for fringe pattern filtering and normalization," *Appl. Opt.* 60, 2022-2036 (2021).

**[16]** Cywińska, Maria & Brzeski, Filip & Krajnik, Wiktor & Patorski, Krzysztof & Zuo, Chao & Trusiak, Maciej. (2021). DeepDensity: Convolutional neural network based estimation of local fringe pattern density. *Optics and Lasers in Engineering*. 145. 106675. 10.1016/j.optlaseng.2021.106675.

**[17]** Yao, Pengcheng & Gai, Shaoyan & Da, Feipeng. (2021). Coding-Net: A multi-purpose neural network for Fringe Projection Profilometry. *Optics Communications*. 489. 126887. 10.1016/j.optcom.2021.126887.

**[18]** Yandong Gao, Guanghui Wang, Geng Wang, Tao Li, Shubi Zhang, Shijin Li, Yansuo Zhang, Tao Zhang, Two-dimensional phase unwrapping method using a refined D-LinkNet-based unscented Kalman filter, *Optics and Lasers in Engineering*, Volume 152, 2022.

**[19]** Javier Gurrola-Ramos, Oscar Dalmau, Teresa Alarcón, U-Net based neural network for fringe pattern denoising, *Optics and Lasers in Engineering*, Volume 149, 2022.

**[20]** Hieu Nguyen, Erin Novak, Zhaoyang Wang, Accurate 3D reconstruction via fringe-to-phase network, *Measurement*, Volume 190, 2022.

**[21]** Yue Sun, Yinxu Bian, Hua Shen, Rihong Zhu, High-accuracy simultaneous phase extraction and unwrapping method for single interferogram based on convolutional neural network, *Optics and Lasers in Engineering*, Volume 151, 2022.

**[22]** Tao Bo, Wang Yan, Qian Xinbo, Tong Xiliang, He Fuqiang, Yao Weiping, Chen Bin, Chen Baojia, Photoelastic Stress Field Recovery Using Deep Convolutional Neural Network, *Frontiers in Bioengineering and Biotechnology*, Volume 10, 2022.

**[23]** K. Ramesh, Sachin Sasikumar, Digital photoelasticity: Recent developments and diverse applications, *Optics and Lasers in Engineering*, Volume 135, 2020, 106186, ISSN 0143-8166.

**[24]** W. Shang, H. Li, J. Liu, and J. Liu, "Algorithm and its application for automatic measurement of the full-field isoclinic parameter by digital phase-shifting photoelasticity," *Appl. Opt.* 61, 10433-10438 (2022).

**[25]** Briñez-de León, J. C., Branch-Bedoya, J. W., and Restrepo-Martínez, A., "Toward photoelastic sensors: a hybrid proposal for imaging the stress field through load stepping methods" in *OSA Imaging and Applied Optics Congress*.

[26] Rodríguez-Marmolejo, Héctor Ulises; Muñoz-Jiménez, Yuliana del R.; Rodríguez, Ulises Mateo; Rodríguez, Saulo Mauricio Demodulación de Patrón de Franjas Cerradas por Corrimiento de Fase Conciencia Tecnológica, núm. 53, 2017 Instituto Tecnológico de Aguascalientes, México.

[27] Qinnan Zhang, Shengyu Lu, Jiaosheng Li, Dong Li, Xiaoxu Lu, Liyun Zhong, Jindong Tian, Phase-shifting interferometry from single frame in-line interferogram using deep learning phase-shifting technology, Optics Communications, Volume 498, 2021, 127226, ISSN 0030-4018, <https://doi.org/10.1016/j.optcom.2021.127226>.

[28] H. Fandiño-Toro, Y. Aristizábal-López, A. Restrepo-Martínez, and J. Briñez-de León, "Fringe pattern analysis to evaluate light sources and sensors in digital photoelasticity," Appl. Opt. 62, 2560-2568 (2023).

[29] Briñez de León, Juan & Restrepo-Martinez, Alejandro & Branch, John. (2018). Computational hybrid phase shifting technique applied to digital photoelasticity. Optik - International Journal for Light and Electron Optics. 157. 287-297. 10.1016/j.ijleo.2017.11.060.

[30] Quiroga, J. and Gonzalez-Cano, A., "Separation of isoclinics and isochromatics from photoelastic data with a regularized phase-tracking technique" Applied optics 39(17), 2931-2940 (2000).

[31] Pandey, A. and Ramesh, K., "Development of a new normalization technique for twelve fringe photoelasticity (tfp)" in [Advancement of Optical Methods & Digital Image Correlation in Experimental Mechanics, Volume 3], 177-180, Springer (2019).

[32] Juan Carlos Briñez-de León, Alejandro Restrepo-Martínez, John W. Branch-Bedoya, Computational analysis of Bayer colour filter arrays and demosaicking algorithms in digital photoelasticity, Optics and Lasers in Engineering, Volume 122, 2019, Pages 195-208, ISSN 0143-8166, <https://doi.org/10.1016/j.optlaseng.2019.06.004>.

[33] Briñez-de León, J. C., Rico-García, M., Restrepo-Martínez, A., and Branch-Bedoya, J. W., "Isochromatic-art: a computational dataset for evaluating the stress distribution of loaded bodies by digital photoelasticity." Mendeley Data, v4 (2020).

[34] Gurton, Kristan & Yuffa, Alex & Videen, Gorden. (2014). Enhanced facial recognition for thermal imagery using polarimetric imaging. Optics letters. 39. 3857-3859. 10.1364/OL.39.003857.

[35] Chang, Jintao & He, Honghui & He, Chao & Ma, Hui. (2016). DoFP polarimeter based polarization microscope for biomedical applications. 97070W. 10.1117/12.2210896.

[36] Li Xiaobo, Han Yilin, Wang Hongyuan, Liu Tiegen, Chen Shih-Chi, Hu Haofeng. Polarimetric Imaging Through Scattering Media: A Review. Frontiers in Physics. Volume 10. 2022. 10.3389/fphy.2022.815296. 2296-424X.

[37] Timoshenko, S. and Goodier, J.N. (1951) Theory of Elasticity. 2nd Edition, McGraw-Hill, New York.

[38] Kramer, Sharlotte Lorraine Bolyard (2009) *Phase-Shifting Full-Field Interferometric Methods for In-Plane Tensorial Stress Determination for Fracture Studies*. Dissertation (Ph.D.), California Institute of Technology. doi:10.7907/M9NV-T722.

[39] Huaizi Tang, Jiuzhou Huang, Jinyuan He, Wen Hua, Shiming Dong, Stress intensity factors for a centrally cracked Brazilian disk under non-uniformly distributed pressure, *Theoretical and Applied Fracture Mechanics*, Volume 114, 2021, 103001, ISSN 0167-8442.

[40] Ramesh, K., Shins, K. Stress field equations for a disk subjected to self-equilibrated arbitrary loads: revisited. *Granular Matter* 24, 49 (2022).

[41] Olaf Ronneberger, Philipp Fischer, Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *ArXiv. abs/1505.04597*. 2015.

[42] W. Zhao, G. Zhang, and J. Li, "Accuracy improvement of demodulating the stress field with StressUnet in photoelasticity," *Appl. Opt.* 61, 8678-8687 (2022).

[43] J. Zhao, L. Liu, T. Wang, X. Wang, X. Du, R. Hao, J. Liu, Y. Liu, and J. Zhang, "VDE-Net: a two-stage deep learning method for phase unwrapping," *Opt. Express* 30, 39794-39815 (2022).

[44] Kaiqiang Wang, Qian Kemao, Jianglei Di, Jianlin Zhao, "Deep learning spatial phase unwrapping: a comparative review," *Adv. Photon. Nexus* 1(1) 014001 (3 August 2022)

[45] Zuo, C., Qian, J., Feng, S. et al. Deep learning in optical metrology: a review. *Light Sci Appl* 11, 39 (2022).

[46] Park, Seonghwan & Kim, Youhyun & Moon, Inkyu. (2021). Automated phase unwrapping in digital holography with deep learning. *Biomedical Optics Express*. 12. 10.1364/BOE.440338.

[47] Mohamad Zaim Awang Pon, & Krishna Prakash K. (2021). Hyperparameter Tuning of Deep learning Models in Keras. *Sparklinglight Transactions on Artificial Intelligence and Quantum Computing (STAIQC)*, 1(1), 36–40.

[48] Hertel, Lars & Collado, Julian & Sadowski, Peter & Ott, Jordan & Baldi, Pierre. (2020). Sherpa: Robust hyperparameter optimization for machine learning. *SoftwareX*. 12. 100591. 10.1016/j.softx.2020.100591.

[49] A. Mittal, A. K. Moorthy and A. C. Bovik, "No-Reference Image Quality Assessment in the Spatial Domain," in *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695-4708, Dec. 2012, doi: 10.1109/TIP.2012.2214050.

[50] W. Zhang, K. Ma, J. Yan, D. Deng, and Z. Wang, "Blind Image Quality Assessment Using a Deep Bilinear Convolutional Neural Network," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 1, pp. 36-47, Jan. 2020, doi: 10.1109/TCSVT.2018.2886771.

[51] Talebi, Hossein & Milanfar, Peyman. (2017). NIMA: Neural Image Assessment. *IEEE Transactions on Image Processing*. PP. 10.1109/TIP.2018.2831899.

[52] Chao Ma, Chih-Yuan Yang, Xiaokang Yang, Ming-Hsuan Yang, Learning a no-reference quality metric for single-image super-resolution, Computer Vision and Image Understanding, Volume 158, 2017, Pages 1-16, ISSN 1077-3142.

[53] H. Zheng, H. Yang, J. Fu, Z. Zha and J. Luo, "Learning Conditional Knowledge Distillation for Degraded-Reference Image Quality Assessment," in 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021 pp. 10222-10231. doi: 10.1109/ICCV48922.2021.01008

[54] K. Ding, K. Ma, S. Wang and E. P. Simoncelli, "Image Quality Assessment: Unifying Structure and Texture Similarity," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 5, pp. 2567-2581, 1 May 2022, doi: 10.1109/TPAMI.2020.3045810.

[55] R. Zhang, P. Isola, A. Efros, E. Shechtman and O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018 pp. 586-595. doi: 10.1109/CVPR.2018.00068

[56] Brieva Sarmiento, María Angélica, (2011). Estudio de tensiones residuales mediante fotoelasticidad en piezas de poliestireno cristal moldeadas por inyección. Universidad de Cartagena. Facultad de Ingeniería.

[57] BFS-U3-51S5PC-C USB 3.1 Blackfly® S, polarized color camera. Edmund Optics Worldwide. Available at: <https://www.edmundoptics.com/p/bfs-u3-51s5pc-c-usb3-blackfly-s-polarized-color-camera/41901/> (Accessed: 04 July 2023).

[58] VCXU-50MP. Industrial cameras / CX series | Baumer USA. Available at: <https://www.baumer.com/us/en/p/38833> (Accessed: 04 July 2023).

[59] C2410Y/Z CMOS 5 MP with Polarization Filters. GigE Vision® with Power over Ethernet (PoE) (pp. 1–2). (2019). 6421 Congress Ave., Boca Raton, FL 33487, USA: IMPERX Industrial Cameras & Imaging Systems. Recuperado de [https://www.cosyco.de/kameras/flaechenkameras/imperx/Datenblaetter\\_20200923/cheeta\\_h\\_c2410YZ\\_poe\\_r2\\_2019\\_w.pdf](https://www.cosyco.de/kameras/flaechenkameras/imperx/Datenblaetter_20200923/cheeta_h_c2410YZ_poe_r2_2019_w.pdf)

[60] Rohatgi A: WebPlotDigitizer. 2019. Available at: <https://automeris.io/WebPlotDigitizer/images/wpd.png>. Accessed July 4, 2023.

[61] Junior, Pedro & Magalhães, Cristina & Neto, Perrin. (2012). New Numerical Method for the Photoelastic Technique. Journal of the Brazilian Society of Mechanical Sciences and Engineering. 34. 531-542. 10.1590/S1678-58782012000400015.

[62] Tokovyy, Yuriy & Hung, K.-M & Ma, Chien-Ching. (2010). Determination of stresses and displacements in a thin annular disk subjected to diametral compression. Journal of Mathematical Sciences. 165. 342-354. 10.1007/s10958-010-9803-6.

[63] Vedel, Mathieu & Lechocinski, Nicolas & Breugnot, Sebastien. (2010). Paper Title: Compact and robust Linear Stokes polarization camera -Description and Applications. EPJ Web of Conferences. 5. 01005. 10.1051/epjconf/20100501005.