Detección de glaucoma mediante redes neuronales a partir de imágenes de fondo de ojo

Sofie Filiaev Nilsen, Stivan Svetoslavov Tinchev, Mario Dorado

Perera 7 de mayo de 2026

Resumen

El glaucoma es una enfermedad ocular crónica caracterizada por un daño pro gresivo del nervio óptico y constituye una de las principales causas de ceguera irre versible a nivel mundial. Su detección temprana resulta especialmente importante porque, en muchas ocasiones, la enfermedad avanza sin síntomas evidentes hasta fases en las que la pérdida visual ya es significativa. En este trabajo se estudia un problema de visión artificial aplicado al ámbito médico: la detección automática de glaucoma a partir de retinografías centradas en la cabeza del nervio óptico. Para ello se emplea un conjunto de imágenes de fondo de ojo dividido en dos clases, *glaucoma* y *non_glaucoma*, y se comparan distintos enfoques basados en redes neuronales: una CNN con clasificador MLP, una CNN con clasificador KAN, autoencoders con espacios latentes de dimensión 16 y 32 combinados con clasificadores MLP y KAN, y un modelo avanzado basado en *transfer learning* con EfficientNetB2. El objeti vo principal es analizar si estos modelos son capaces de aprender patrones visuales asociados al glaucoma y comparar el efecto de diferentes estrategias de extracción y clasificación de características.

Palabras clave: glaucoma, fondo de ojo, redes neuronales convolucionales, autoencoder, KAN, EfficientNet, aprendizaje profundo.

1 Introducción

El glaucoma es un grupo de enfermedades oculares neurodegenerativas que producen un daño progresivo en el nervio óptico. Este daño suele asociarse a la presión intraocular

elevada, aunque también puede aparecer en pacientes con valores de presión considerados normales. Su importancia clínica se debe a que la pérdida de fibras nerviosas y de campo visual es irreversible: una vez que el paciente percibe síntomas claros, el deterioro puede encontrarse ya en una fase avanzada.

A escala mundial, el glaucoma representa una de las principales causas de ceguera irre versible. Tham et al. (2014) estimaron que el número de personas afectadas crecería de forma considerable en las décadas siguientes, con una proyección de 111,8 millones de casos para 2040. Esta carga sanitaria hace que la detección temprana sea una prioridad, especialmente en contextos donde el acceso a especialistas es limitado o donde el cribado poblacional puede resultar costoso.

Desde el punto de vista clínico, la evaluación del glaucoma se apoya en distintas pruebas: medición de la presión intraocular, exploración del campo visual, tomografía de coherencia óptica y análisis del fondo de ojo. En las retinografías, los especialistas observan la cabeza del nervio óptico y valoran signos como el aumento de la excavación papilar, la relación copa-disco, el adelgazamiento del anillo neurorretiniano, la presencia de hemorragias pe ripapilares y los defectos en la capa de fibras nerviosas de la retina. Estos criterios son relevantes porque el glaucoma modifica progresivamente la estructura del nervio óptico antes de que el paciente note una pérdida visual evidente (Fingeret et al., 2005).

Figura 1: Comparación entre un ojo sano y un ojo con glaucoma

Figura 2: Efecto del Glaucoma sobre la visión

2 Motivación del uso de redes neuronales

La detección automática de glaucoma mediante aprendizaje profundo surge como una herramienta de apoyo al diagnóstico, no como sustitución del criterio médico. Las re des neuronales convolucionales son especialmente adecuadas para este tipo de problema porque pueden aprender patrones espaciales complejos directamente desde imágenes. En lugar de depender únicamente de medidas manuales, una CNN puede extraer bordes, tex turas, contrastes, estructuras vasculares y configuraciones del disco óptico que resultan útiles para diferenciar ojos sanos de ojos con signos compatibles con glaucoma.

Varios estudios han mostrado el potencial de los modelos de aprendizaje profundo en re tinografías. Ahn et al. (2018) entrenaron modelos de clasificación para detectar glaucoma avanzado y temprano usando fotografías de fondo de ojo, mostrando que una CNN podía alcanzar resultados competitivos frente a enfoques clásicos. De forma similar, Christopher et al. (2018) evaluaron arquitecturas profundas y transferencia de aprendizaje para de tectar neuropatía óptica glaucomatosa, observando que el *transfer learning* podía ser útil cuando se trabaja con imágenes médicas. Liu et al. (2019) desarrollaron y validaron un sistema de aprendizaje profundo para detectar neuropatía óptica glaucomatosa a partir de fotografías de fondo de ojo, subrayando además la utilidad potencial de estos sistemas en programas de cribado. Otros trabajos, como el de Shibata et al. (2018), emplearon redes residuales para el cribado del glaucoma mediante retinografías.

En este contexto, el presente trabajo compara varias familias de modelos. Por una parte, se evalúan arquitecturas CNN entrenadas desde cero con dos tipos de clasificadores finales: una MLP tradicional y una red KAN. Por otra parte, se estudia el uso de autoencoders como método de preentrenamiento y reducción de dimensionalidad. Finalmente, se incor pora un enfoque avanzado basado en EfficientNetB2, una arquitectura eficiente diseñada mediante escalado compuesto de profundidad, anchura y resolución (Tan and Le, 2019).

3 Datos y preprocesado

El conjunto de datos utilizado procede de la ruta:

/kaggle/input/datasets/jeremypoveda/onh-fundus-images-for-glaucoma/

Dataset-Classifier/

La base de datos está organizada en dos divisiones principales: train, empleada para el entrenamiento, y test, usada como conjunto de validación o prueba durante los expe rimentos. Las imágenes pertenecen a dos clases: glaucoma y non_glaucoma. Todas las

imágenes son retinografías en color centradas en la cabeza del nervio óptico, una región anatómica especialmente relevante porque en ella se manifiestan signos estructurales aso ciados al glaucoma.

Con el objetivo de comparar los modelos bajo condiciones homogéneas, se aplicó un pre procesado común en todos los experimentos principales:

- tamaño de entrada: 150 × 150 píxeles;
- modo de color: RGB, con tres canales;
- tamaño de lote: 32 imágenes;
- etiquetas enteras con clasificación binaria;
- función de pérdida para clasificación: sparse_categorical_crossentropy; •

métrica principal: accuracy.

El redimensionamiento a 150 × 150 permite que todas las imágenes tengan una forma fija y reduce el coste computacional, algo importante al entrenar varios modelos en Kaggle. Mantener el formato RGB conserva la información cromática de las retinografías, que puede ser relevante para distinguir estructuras del disco óptico, vasos sanguíneos, zonas de atrofia y variaciones de iluminación.

4 Modelos evaluados

4.1 Modelo 1: CNN con clasificador MLP

El primer modelo utiliza una red neuronal convolucional como extractor de características y una red totalmente conectada como clasificador final. La entrada del modelo tiene dimensión 150×150×3. Antes de la parte convolucional se aplica aumento de datos para mejorar la capacidad de generalización:

• volteo horizontal y vertical aleatorio: RandomFlip("horizontal_and_vertical"); • rotación aleatoria con factor 0.2;

• zoom aleatorio con factor 0.1;

• reescalado de píxeles mediante Rescaling(1./255).

El extractor CNN está formado por cuatro bloques convolucionales. Cada bloque contiene una convolución 3 × 3 con activación ReLU, seguida de normalización por lotes y una operación de *max pooling* 2 × 2. El número de filtros aumenta progresivamente: 32, 64, 128 y 256. Esta estructura permite que las primeras capas capturen patrones simples, como bordes y contrastes, mientras que las capas más profundas aprenden características más abstractas relacionadas con la morfología del nervio óptico.

Tras la parte convolucional, se aplana la representación mediante Flatten. El clasificador final es una MLP con una capa densa de 512 neuronas y activación ReLU, seguida de una capa Dropout con tasa 0.5. La salida está formada por dos neuronas con activación softmax, correspondientes a las clases glaucoma y non_glaucoma.

El modelo se entrenó con el optimizador Adam, una tasa de aprendizaje de 10^{-4}, hasta un máximo de 100 épocas. Para evitar sobreajuste se empleó *early stopping* monitorizando val_loss, con paciencia 12 y restauración de los mejores pesos.

Cuadro 1: Resumen del modelo CNN + MLP.

Componente	Configuración
Entrada	150 × 150 × 3 RGB
Aumento de datos	Flip H/V, rotación 0.2, zoom 0.1
Bloques CNN	Conv2D 32, 64, 128, 256; kernel 3 × 3
Normalización	Batch normalization en cada bloque
Pooling	MaxPooling2D 2 × 2
Clasificador	Dense 512 ReLU + Dropout 0.5
Salida	Dense 2 Softmax
Optimizador	Adam, learning rate 10^{-4}
Épocas máximas	100
Early stopping	val_loss, paciencia 12

4.2 Modelo 2: CNN con clasificador KAN

El segundo modelo mantiene exactamente el mismo extractor convolucional que el modelo anterior: entrada 150 × 150 × 3, aumento de datos, reescalado, cuatro bloques convolucio nales con filtros 32, 64, 128 y 256, normalización por lotes y *max pooling*. La diferencia se encuentra en el clasificador final.

En lugar de utilizar una MLP con una capa densa de 512 neuronas, este modelo incorpora una red KAN personalizada. Las redes KAN, o *Kolmogorov-Arnold Networks*, han sido propuestas como alternativa a las MLP tradicionales. Mientras que una MLP aprende

pesos lineales y aplica funciones de activación fijas en las neuronas, una KAN desplaza parte de la capacidad expresiva hacia funciones aprendibles asociadas a las conexiones (Liu et al., 2024).

En este caso, la implementación usa una aproximación polinómica estabilizada. La capa definida como CapaKAN recibe la representación aplanada procedente de la CNN y genera 64 unidades. El grado polinómico empleado es 3. Para evitar inestabilidades numéricas, la capa incluye normalización mediante LayerNormalization, recorte de valores entre -2 y 2, e inicialización de pesos con una distribución normal de desviación típica 0.01. Después de la capa KAN se aplica BatchNormalization, Dropout(0.5) y una salida Dense(2, softmax).

El entrenamiento se configuró igual que en el modelo CNN + MLP: optimizador Adam, tasa de aprendizaje 10^{-4}, máximo de 100 épocas y *early stopping* sobre val_loss con paciencia 12.

Cuadro 2: Resumen del modelo CNN + KAN.

Componente	Configuración
Extractor CNN	Igual que CNN + MLP
Representación	Flatten
Capa KAN	64 unidades, grado 3
Estabilización KAN	LayerNormalization, clipping [-2, 2]
Regularización	Dropout 0.5
Salida	Dense 2 Softmax
Optimizador	Adam, learning rate 10^{-4}
Épocas máximas	100

Early stopping val_loss, paciencia 12

4.3 Modelos 3 y 4: autoencoders con espacios latentes de dimen sión 16 y 32

Los modelos tercero y cuarto se basan en una estrategia de preentrenamiento no super visado mediante autoencoders. Un autoencoder aprende a reconstruir su propia entrada, obligando a que la información de la imagen pase por un espacio latente comprimido. En este trabajo se evaluaron dos tamaños de espacio latente: 16 y 32 dimensiones.

La arquitectura del encoder comienza con una imagen de entrada 150×150×3. Se aplican tres bloques convolucionales:

- Conv2D(32, 3x3, relu) + MaxPooling2D;
- Conv2D(64, 3x3, relu) + MaxPooling2D;
- Conv2D(128, 3x3, relu) + MaxPooling2D.

Después, la salida se aplana con Flatten y se proyecta al espacio latente mediante una capa Dense(dim_latente, relu), donde dim_latente toma los valores 16 y 32.

El decoder recibe el vector latente y trata de reconstruir la imagen original. Primero se aplica una capa Dense(41472, relu), seguida de Reshape(18, 18, 128). A continua ción, se usan convoluciones y operaciones de sobremuestreo:

- Conv2D(128, relu) + UpSampling2D;
- Conv2D(64, relu) + UpSampling2D;
- Conv2D(32, relu) + UpSampling2D;
- Conv2D(3, sigmoid);
- redimensionamiento final a 150 × 150.

El autoencoder se entrenó con el optimizador Adam y pérdida de error cuadrático medio, mse. El número máximo de épocas fue 50 y se aplicó *early stopping* sobre val_loss con paciencia 10. Una vez entrenado, el encoder se guardó y se congeló para utilizarlo como extractor de características. Sobre ese vector latente se

entrenaron dos clasificadores: una MLP y una KAN.

El clasificador MLP aplicado al espacio latente contiene BatchNormalization, una capa densa de 128 neuronas con activación swish, Dropout(0.40), una capa densa de 64 neuronas con activación swish, Dropout(0.30) y salida softmax.

El clasificador KAN aplicado al espacio latente utiliza dos capas KAN con B-splines. La primera tiene 64 unidades y la segunda 32. En ambas se emplea grid_size=5 y spline_order=3. Entre ellas se aplican tasas de dropout de 0.35 y 0.25. A diferencia de la KAN del modelo 2, aquí la capa implementa una base B-spline sobre una rejilla extendida entre -1 y 1, combinando una parte base y una parte spline.

Los clasificadores se entrenaron con Adam, tasa de aprendizaje 10^{-3}, pérdida de entropía cruzada categórica dispersa y máximo de 40 épocas. Se emplearon *early stopping*, reduc ción automática de la tasa de aprendizaje, guardado del mejor modelo y registro CSV del entrenamiento. El *early stopping* monitorizó val_accuracy con paciencia 10, mientras que la reducción de tasa de aprendizaje usó factor 0.5, paciencia 4 y tasa mínima 10^{-6}.

Cuadro 3: Resumen de los modelos basados en autoencoder.

Componente Configuración
Latentes evaluados 16 y 32 dimensiones
Encoder Conv2D 32, 64, 128 + MaxPooling
Proyección latente Dense 16 o Dense 32, ReLU
Decoder Dense 41472, Reshape 18x18x128, Conv2D + UpSampling Salida decoder
Conv2D 3 sigmoid + Resizing 150x150
Loss autoencoder MSE
Épocas autoencoder 50, early stopping paciencia 10
Clasificador MLP BN, Dense 128 swish, Dropout 0.40, Dense 64 swish, Dro pout 0.30
Clasificador KAN KAN 64 + Dropout 0.35, KAN 32 + Dropout 0.25 KAN
B-spline grid_size=5, spline_order=3
LR clasificadores 10^{-3}
Épocas clasificadores 40, early stopping paciencia 10

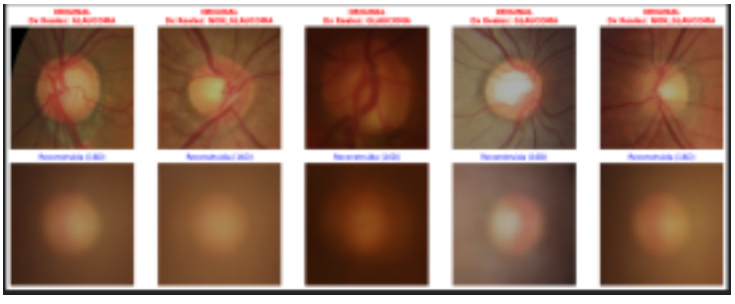

Figura 3: Reconstrucción de imagenes mediante Autoencoders con espacio latente de 16 dimensiones

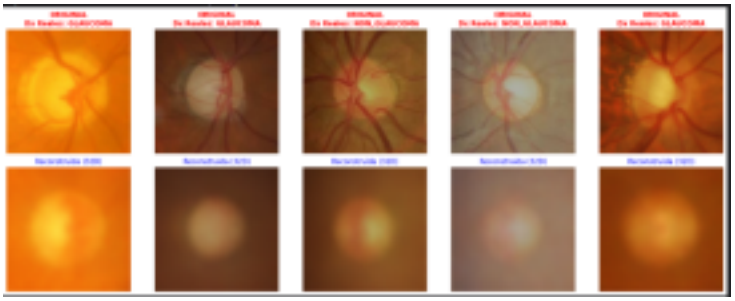

Figura 4: Reconstrucción de imagenes mediante Autoencoders con espacio latente de 32 dimensiones

4.4 Modelo 5: EfficientNetB2 con transferencia de aprendizaje

El último modelo introduce una estrategia más avanzada basada en transferencia de apren dizaje. En lugar de entrenar desde cero todo el extractor visual, se utiliza EfficientNetB2 como red base. EfficientNet fue propuesta por Tan and Le (2019) como una familia de re des convolucionales escaladas de forma equilibrada en profundidad, anchura y resolución, consiguiendo una buena relación entre precisión y coste computacional.

El modelo trabaja con dos versiones de cada imagen: la imagen original y una imagen procesada. La versión procesada se obtiene aumentando el contraste con factor 1.35, apli cando una corrección gamma de 0.9 y aumentando la saturación con factor 1.10. Después se recortan los valores para mantenerlos dentro del rango válido. Esta estrategia busca resaltar diferencias visuales que podrían ser relevantes en el nervio óptico y en la retina.

Además, se aplica aumento de datos durante el entrenamiento:

- volteo horizontal y vertical;
- rotación aleatoria 0.10;
- zoom aleatorio 0.10;
- contraste aleatorio 0.15;

• traslación aleatoria 0.04 en altura y anchura.

La arquitectura tiene dos ramas de entrada: original y processed. Ambas pasan por el mismo backbone EfficientNetB2, configurado sin la capa superior de clasificación. Si es posible, se cargan pesos preentrenados en ImageNet; si no, el código continúa con pesos aleatorios. Tras el backbone, cada rama pasa por GlobalAveragePooling2D. Las dos re presentaciones se concatenan y se envían a un clasificador MLP con BatchNormalization, una capa densa de 384 neuronas con activación swish, Dropout(0.35), una capa densa de 128 neuronas con activación swish, Dropout(0.20) y salida softmax.

El entrenamiento se divide en dos fases. En la primera fase, EfficientNetB2 permanece congelada y solo se entrena el clasificador final. Se usa Adam con tasa de aprendizaje 3×10^{-4} durante un máximo de 6 épocas, con *early stopping* de paciencia 3. En la segunda fase, se descongela el último 35% de las capas del backbone, manteniendo congeladas las capas de BatchNormalization. Esta fase se entrena con Adam y tasa de aprendizaje 10^{-5} durante un máximo de 14 épocas, con paciencia 4. Esta reducción de la tasa de aprendizaje es importante para ajustar suavemente los pesos preentrenados sin destruir las representaciones aprendidas previamente.

La evaluación final incluye *test-time augmentation*. El modelo predice usando cuatro trans formaciones: imagen original, volteo horizontal, volteo vertical y rotación de 90 grados. Las probabilidades obtenidas se promedian para calcular la predicción final. Además de la exactitud, el código calcula precisión, sensibilidad, especificidad y la matriz de conteos TP, TN, FP y FN.

Cuadro 4: Resumen del modelo avanzado con EfficientNetB2.

Componente	Configuración
Backbone	EfficientNetB2 sin capa superior
Pesos	ImageNet si están disponibles; si no, aleatorios
Entradas	Imagen original + imagen procesada
Procesado	Contraste 1.35, gamma 0.9, saturación 1.10
Aumento de datos	Flip H/V, rotación, zoom, contraste, traslación
Pooling	GlobalAveragePooling2D por rama
Fusión	Concatenación de características
Clasificador	Dense 384 swish, Dropout 0.35, Dense 128 swish, Dropout 0.20
Fase 1	Backbone congelado, LR 3×10^{-4}, 6 épocas

Fase 2 Último 35% descongelado, LR 10^{-5}, 14 épocas Evaluación TTA: identidad, flip H, flip V, rotación 90 grados Métricas Accuracy, precision, sensitivity, specificity, TP, TN, FP, FN

5 Comparación experimental

La Tabla 5 muestra algunos de los resultados tras ejecutar varias veces los códigos co rrespondientes a cada implementación.También se incluirá la época en la que se detiene cada método tra no haber ninguna mejora en la *LOSS*. En el caso del quinto modelo, se icnluye las épocas para cada fase.

Cuadro 5: Comparación de resultados de los modelos evaluados.

Modelo Val. acc. Val. loss Épocas Observación

CNN + MLP 0.8631 0.3606 40 Modelo base denso CNN + KAN 0.8759 0.3087 31 Sustituye MLP por KAN AE 16D + MLP 0.7517 0.5139 34 Latente compacto AE 16D + KAN 0.7371 0.5472 11 KAN sobre latente 16D AE 32D + MLP 0.6888 0.5852 30 Mejor reconstrucción esperada AE 32D + KAN 0.7293 0.5556 37 KAN sobre latente 32D EfficientNetB2 0.8629 0.3604 6/11 Transfer learning + TTA

A partir de las observaciones realizadas durante el trabajo, los modelos basados en auto encoder muestran una diferencia interesante entre calidad de reconstrucción y rendimiento de clasificación. Un espacio latente de 32 dimensiones conserva más información visual y, por tanto, permite reconstrucciones más fieles. Sin embargo, una mayor capacidad la tente no implica necesariamente una mejor separación entre clases,como en nuestro caso. Tanto el Autoencoder con un espacio latente de 16 dimensiones con clasifiación basada en una MLP como en una KAN superan en

acierto a sus versiones con espacio latente de dimensión 32.

La comparación entre MLP y KAN también resulta relevante. Si comparamos la clasifi cación realizada por las dos redes neuroanles tras el tratamiento de la imagen tanto con CNN y Autoencoders,vemos que la exactitud obtenida por las KANs es algo superior frente a las MLPs, salvo para el Autoencoder con espacio latente de dimensión 16, donde se obtiene un accuracy del 73.71% frente a un 75.17% respectivamente.

Finalmente,a pesar de las técnicas avanzadas implementadas por EfficientNetB2 y obtener una exactitud del 86.29%, es ligeramente superado por las dos redes CNN, siendo la mejor clasificación la realizada por la KAN, con un 87.59%

6 Conclusiones

En este trabajo se ha abordado la detección automática de glaucoma a partir de imágenes de fondo de ojo mediante diferentes arquitecturas de redes neuronales. En primer lugar, se entrenaron modelos CNN desde cero con clasificadores MLP y KAN. Posteriormente, se exploró el uso de autoencoders como técnica de preentrenamiento y reducción de dimensio nalidad, comparando espacios latentes de 16 y 32 dimensiones. Por último, se implementó un enfoque avanzado basado en EfficientNetB2 y transferencia de aprendizaje.

El estudio permite comprobar que las redes neuronales pueden aprender patrones visuales asociados al glaucoma en retinografías centradas en el nervio óptico. Sin embargo, tam bién muestra que aumentar la complejidad del modelo no siempre garantiza una mejora clara en los datos de prueba,como en el caso de los Autoencoders. En tareas médicas, es pecialmente con conjuntos de datos limitados, la capacidad de generalización depende de un equilibrio entre arquitectura, regularización, preprocesado, aumento de datos y calidad de las etiquetas.

Como trabajo futuro, sería recomendable incorporar métricas clínicas adicionales, como sensibilidad y especificidad para todos los modelos, analizar matrices de confusión, usar validación cruzada y probar técnicas de interpretabilidad visual como mapas de activación. Esto permitiría evaluar no solo si el modelo acierta, sino también si toma sus decisiones atendiendo a regiones clínicamente relevantes del nervio óptico.

Referencias

Ahn, J. M., Kim, S., Ahn, K.-S., Cho, S.-H., Lee, K. B., and Kim, U. S. (2018). A deep learning model for the detection of both advanced and early glaucoma using fundus photography. *PLOS ONE*, 13(11):e0207982.

Christopher, M., Belghith, A., Bowd, C., Proudfoot, J. A., Goldbaum, M. H., Weinreb, R. N., Girkin, C. A., Liebmann, J. M., and Zangwill, L. M. (2018). Performance of deep learning architectures and transfer learning for detecting glaucomatous optic neuropathy in fundus photographs. *Scientific Reports*, 8:16685.

Fingeret, M., Medeiros, F. A., Susanna, R., and Weinreb, R. N. (2005). Five rules to eva luate the optic disc and retinal nerve fiber layer for glaucoma. *Optometry*,

76(11):661– 668.

Liu, H., Li, L., Wormstone, I. M., Qiao, C., Zhang, C., Liu, P., Li, S., Wang, H., Mou, D., Pang, R., Yang, D., Zangwill, L. M., Moghimi, S., Hou, H., Bowd, C., Jiang, L., Chen, Y., Hu, M., Xu, Y., Kang, H., Ji, X., Chang, R. T., Tham, C. C., Cheung, C. Y., Ting, D. S. W., Wong, T. Y., Wang, Z., Weinreb, R. N., Xu, M., and Wang, N. (2019). Development and validation of a deep learning system to detect glaucomatous optic neuropathy using fundus photographs. *JAMA Ophthalmology*, 137(12):1353–1360.

Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačič, M., Hou, T. Y., and Tegmark, M. (2024). KAN: Kolmogorov-arnold networks.

Shibata, N., Tanito, M., Mitsuhashi, K., Fujino, Y., Matsuura, M., Murata, H., and Asao ka, R. (2018). Development of a deep residual learning algorithm to screen for glaucoma from fundus photography. *Scientific Reports*, 8:14665.

Tan, M. and Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutio nal neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6105–6114.

Tham, Y.-C., Li, X., Wong, T. Y., Quigley, H. A., Aung, T., and Cheng, C.-Y. (2014). Global prevalence of glaucoma and projections of glaucoma burden through 2040: A systematic review and meta-analysis. *Ophthalmology*, 121(11):2081–2090.