Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Matemáticas y Física

Maestría en Ciencia de Datos

Implementación de Redes Neuronales Convolucionales para el reconocimiento de imágenes de objetos perdidos en Aeropuertos

---

TRABAJO RECEPCIONAL que para obtener el **GRADO** de
Maestro en Ciencia de Datos

Presenta:
Jesús Alvarez Castellanos

Director:
Mtro. Arturo Silva Gálvez

Tlaquepaque, Jalisco, 29 de mayo de 2026

Implementación de Redes Neuronales Convolucionales para el reconocimiento de imágenes de objetos perdidos en Aeropuertos

Jesús Alvarez Castellanos

Resumen

El desarrollo acelerado de la tecnología de procesamiento digital de imágenes a nivel global ha logrado obtener grandes avances dentro de la rama de las ciencias computacionales, particularmente en el campo de la visión por computadora y el reconocimiento de patrones. La implementación de una gran cantidad de técnicas de preprocesamiento, junto con arquitecturas de aprendizaje profundo cada vez más sofisticadas, contribuyen en conjunto a un mejor modelado y aprendizaje automático que permita generar estimaciones favorables para la resolución de problemas complejos en diversos dominios de aplicación. En este trabajo se presenta el desarrollo de un modelo de redes neuronales para el reconocimiento y clasificación de imágenes de objetos perdidos en aeropuertos, abordando una problemática recurrente en las instalaciones aeroportuarias donde diariamente se registran numerosos casos de pertenencias extraviadas. El modelo propuesto tiene como objetivo principal procesar las imágenes de objetos extraviados en las instalaciones aeroportuarias para clasificarlos automáticamente que pueda facilitar a los propietarios la recuperación de sus pertenencias de manera eficiente, reduciendo así los tiempos de búsqueda y mejorando la gestión de objetos perdidos.

Se selecciona como base de datos el repositorio ImageNet, creada por la Universidad de Princeton, contiene más de dos millones de imágenes organizadas en categorías semánticas denominadas synsets, derivadas de la estructura jerárquica de WordNet, también un recurso lexico creado en Princeton. Esta ontología organiza conceptos visuales en árboles con hasta catorce niveles de profundidad. Las categorias utilizadas son recaudadas mediante el descarte de objetos no permitidos en salas de espera dentro de los aeropuertos de acuerdo a las normas internacionales de aviación civil. Una vez seleccionadas las clases, cada imagen fue sometida a un pipeline de cinco etapas para su preprocesamiento. La arquitectura principal del proyecto fue la CNN extendida, organizada en seis bloques convolucionales ascendentes de 32 a 512 filtros, incorporando en cada bloque `BatchNormalization`, `Dropout` *y terminando en una capa* `GlobalAveragePooling2D` *antes de la clasificación densa.*

Se entrenan y se evaluan 15 experimentos en los cuales se realizan ajustes sistemáticos a los diferentes hiperparámetros que configuran a la arquitectura del modelo principal. Dichos experimentos se dividen en 3 bloques donde en cada uno de ellos se fueron ajustando distintos tipos de hiperparámetros referentes al optimizados, aumentación de los datos y regularización de las capas convolucionales. Se selecciona el modelo con el mejor rendimiento de acuerdo a la métrica de exactitud.

El modelo generado y seleccionado con mejor rendimiento demostró ser capaz de procesar las imagenes con un pipeline completo de transformaciones para mejorar la representación de los objetos a clasificar. También entregó una clasificación base con gran potencial de mejora de su rendimiento. Se planeá incorporar un mejor repositorio de imagenes que puedan representar de manera más presisa el dominio de categorias propias a los aeropuertos a ser implementados para mejorar el rendimiento del modelo.

Tabla de Contenidos

Índice de figuras

Página

Índice de tablas

Página

Dedicado a mis padres, familia y seres queridos que me han apoyado en el proceso de mis estudios y a lo largo de mi vida. Gracias por motivarme en mejorar y crecer como persona todos los días.

1 Introducción

1.1 Contexto

El crecimiento acelerado de la población a nivel global en el último siglo y el alcance de los 8,000 millones de personas representan una carga de estrés para los individuos. Como señala el Fondo de Población de las Naciones Unidas, este crecimiento acelera la urbanización de las ciudades. Si bien supone un progreso para la estructura socioeconómica, en este contexto genera efectos en la sociedad tales como la aglomeración que dificultan la movilidad y el transporte masivo de la población, entre otros. [1]

La perdida de pertenencias crece constantemente en todos los entornos, tales como los hogares, métodos de transporte, lugares de trabajo y espacios de esparcimiento, debido a la dinámica acelerada y compleja del día a día del ser humano. Aunado a este crecimiento poblacional se encuentra el desarrollo tecnológico acelerado, el cual ha logrado brindar diferentes tipos de alternativas para la solución al problema de los objetos perdidos. Existen dispositivos adjuntos o adheribles a las pertenencias permiten rastrear su ubicación mediante distintas tecnologías como GPS, RF y Bluetooth, entre otras. [2]

No obstante, las tecnologías mencionadas no resuelven la gestión del objeto una vez que este ha sido separado de su dueño y recuperado por un tercero. Estas soluciones son preventivas o de reacción inmediata por parte del propietario y se limitan a aplicaciones de proximidad o geolocalización activa, diseñadas para alertar al usuario si un objeto abandona un área específica o para registrar la última ubicación conocida. [3]

Ante esta situación, la mayoría de las organizaciones tanto privadas como públicas, con instalaciones físicas que cuentan con alto flujo de personas, han buscado implementar módulos para el control de objetos perdidos. Los usuarios de sus instalaciones pueden acudir a entregar objetos olvidados que hayan encontrado dentro de las instalaciones de la organización. Al igual que las personas que por descuido olvidaron dichos objetos puedan acudir a corroborar que sus pertenencias hayan sido entregadas en el módulo, comprobar su

propiedad y reclamarlas, siendo el ITESO una de las organizaciones educativas que ha implementado el uso de este tipo de módulos. [4]

En los módulos mencionados previamente, el proceso de recepción es ejecutado normalmente por una persona que atiende a los usuarios que acuden a realizar la entrega o el reclamo de algún objeto. Se realiza el registro de las recepciones en bitácoras de papel donde se asienta el nombre de la persona que entrega el objeto, un método de contacto, qué objeto se está entregando, así como una breve descripción de este y el lugar y momento en que se encontró. Posteriormente a su registro, se procede a etiquetar con el número de registro de la bitácora y almacenar normalmente junto a los objetos del mismo tipo con la finalidad de agilizar el siguiente proceso. La reclamación es el proceso donde los usuarios acuden al módulo a buscar sus pertenencias extraviadas para que el personal escuche la descripción, cuando y donde ocurrió la pérdida, y así poder mostrar al usuario los objetos que el personal en turno logre identificar coincidentes con la descripción proporcionada para que este pueda indicar cuál es de su pertenencia. El último paso es solicitar un método por el cual se pueda verificar que el usuario es parte de la organización o tuvo acceso a las instalaciones en el momento del abandono.

Este proceso manual ha sido estudiado y se ha buscado implementar sistemas para su administración que faciliten el registro y búsqueda de las bitácoras asegurando que los objetos perdidos sean recuperados de manera efectiva y oportuna [5]. En el ámbito específico de la gestión aeroportuaria, la implementación de un Sistema de Información Gerencial (SIG) orientado a la administración de objetos extraviados permite optimizar el flujo de procesos, transformando procedimientos tradicionales en operaciones automatizadas que garantizan la integridad y disponibilidad de los datos. Esta transición tecnológica no solo simplifica el esfuerzo requerido por el personal de servicio al cliente en las etapas de recepción, almacenamiento y entrega, reduciendo significativamente los tiempos de ejecución y la propensión al error humano inherente a la transcripción de datos en bitácoras físicas, sino que también facilita la generación de reportes gerenciales para la toma de decisiones y asegura el cumplimiento de estándares de calidad y auditoría, contribuyendo directamente al incremento de la satisfacción y lealtad de los usuarios al percibir una mayor eficiencia y transparencia en la recuperación de sus pertenencias. [6]

En este escenario, el aprendizaje profundo (Deep Learning) se erige como una metodología capaz de reducir la dependencia de la labor manual y disminuir significativamente los costos de servicio mediante la construcción de modelos de reconocimiento y clasificación que permiten identificar categorías de objetos de manera rápida y precisa.

Dentro del espectro de las redes neuronales convolucionales (CNN), cuya aplicación se ha extendido ampliamente en diversos subcampos científicos

de reconocimiento de imágenes, destaca la implementación de arquitecturas ligeras como MobileNetV2, la cual ha demostrado ser un ejemplo sobresaliente al sustituir la convolución tradicional por la convolución separable en profundidad (depth-separable convolution), reduciendo drásticamente tanto la cantidad de parámetros como el costo computacional del modelo sin comprometer significativamente la precisión. Para optimizar el entrenamiento de estas redes, es común recurrir a la transferencia de aprendizaje (Transfer Learning), una técnica que aprovecha los pesos preentrenados en conjuntos de datos masivos, como ImageNet, para evitar el entrenamiento desde cero, acelerando así la convergencia de la red y mejorando la capacidad de generalización del modelo ante nuevas tareas de clasificación. [7]

El presente documento se organiza en los siguientes capítulos:

- **Introducción.** *Este capítulo proporciona un trasfondo del problema de pérdida de objetos en aeropuertos, una justificación para el estudio y una descripción general de los objetivos del proyecto.*

- **Metodología.** *Este capítulo proporciona el enfoque metodológico de la investigación, incluyendo una descripción de los datos, el análisis exploratorio y una explicación de los modelos y métricas utilizados.*

- **Resultados.** *Este capítulo presenta y discute los hallazgos del análisis, proporcionando de manera detallada los resultados obtenidos y la comparación de los experimentos realizados.*

- **Conclusiones.** *En este capítulo final se ofrecen conclusiones basadas en los hallazgos de la investigación, así como recomendaciones para futuras investigaciones en este campo.*

1.2 Justificación

Este proyecto surgió a partir de la necesidad de incorporar herramientas tecnológicas al área de Objetos Perdidos en los aeropuertos. Los procedimientos de registro, almacenamiento y recuperación de pertenencias se realizaban de manera manual mediante bitácoras físicas. Esta situación dificultaba la trazabilidad de los objetos y generaba tiempos de respuesta prolongados, lo que afectaba la calidad del servicio brindado a los usuarios que requerían recuperar artículos extraviados durante su tránsito por las instalaciones aeroportuarias. La ausencia de sistemas automatizados limitaba la eficiencia operativa del personal encargado y aumentaba la probabilidad de errores en la documentación y seguimiento de los objetos recopilados.

Una vez que fue solicitado el desarrollo al area de TI una solución capaz de administrar el registro de objetos extraviados y atender solicitudes de búsqueda de forma más eficiente, el cual es desarrollado a la par de este proyecto, surgió la necesidad de integrar un método que permitiera clasificar automáticamente los objetos a partir de una imagen capturada al momento de su recepción. La incorporación de un modelo de reconocimiento visual proporcionaría una segunda fuente de información, complementaria a los registros manuales, permitiendo organizar y recuperar los artículos con mayor rapidez. De esta manera, el sistema no solo favorecería la reducción de tiempos de atención, sino que también contribuiría a mejorar la precisión en la identificación y descripción de los objetos, fortaleciendo la gestión interna del departamento y ofreciendo una experiencia de servicio más satisfactoria para los usuarios.

Cuando un objeto es recibido y registrado en el área de Objetos Perdidos dentro del aeropuerto, se captura una o varias imágenes como parte del proceso de documentación. Dichas imágenes se almacenan junto con información complementaria, como la fecha de recepción, el lugar de hallazgo, la ruta digital de almacenamiento y el testimonio de la persona que entrega el objeto al módulo correspondiente. En este punto, resulta necesario asignar una clasificación adecuada que permita catalogar el artículo dentro de la base de datos y facilitar su localización posterior. La necesidad principal consiste en que el sistema sea capaz de enviar la imagen al modelo de clasificación para que este realice el preprocesamiento y genere la predicción correspondiente, proporcionando la categoría más probable para dicho objeto.

1.3 Problema

Durante la investigación del contexto, se identificó que la clasificación manual que realizaba el personal era un proceso lento, dependiente de la experiencia del operador y susceptible a inconsistencias en la categorización. Esta situación dificultaba la trazabilidad del objeto y prolongaba los tiempos de recuperación cuando los propietarios acudían a solicitar sus pertenencias. Además, las categorías disponibles para la clasificación de los objetos no podían establecerse de forma arbitraria, ya que debían ajustarse a las políticas internas del departamento de Objetos Perdidos, así como a las normas de seguridad aeroportuaria nacionales e internacionales. Dichas normativas determinan qué artículos pueden permanecer en resguardo temporal dentro de las salas de espera y cuáles deben ser restringidos o transferidos a autorización correspondiente.

Debido a que actualmente el sistema informático de Gestión se encuentra en proceso de ser desarrollado, no se cuenta con una base de datos propia con imágenes del departamento de objetos perdidos, por lo que es necesario comenzar el entrenamiento de los modelos con datos disponibles al público. Actualmente existen diferentes repositorios públicos para la investigación sobre visión computacional uno de ellos es ImageNet el cual cuenta con una gran variedad de clases de objetos ligados a WordNet para su clasificación. Esta base de datos se constituye como una ontología de imágenes a gran escala construida sobre la estructura jerárquica de WordNet, buscando poblar la mayoría de los 80,000 "synsets"(conjuntos de sinónimos) con un promedio de 500 a 1000 imágenes limpias y de resolución completa. La elección de ImageNet como fuente primaria para el entrenamiento inicial se justifica no solo por su escala, que supera significativamente a otros conjuntos de datos en términos de cantidad de imágenes y categorías , sino también por la calidad de sus anotaciones, las cuales han sido sometidas a procesos de control de calidad humano para asegurar una alta precisión en el etiquetado. [8]

Sin embargo, la inmensa amplitud de ImageNet, que abarca desde subárboles taxonómicos de mamíferos, anfibios y formaciones geológicas hasta vehículos e instrumentos musicales, presenta un desafío de selección. Dado que el sistema debe responder a las normativas de seguridad y custodia previamente citadas, no todas las categorías disponibles en el repositorio son pertinentes; por ejemplo, categorías biológicas o de grandes vehículos no corresponden a los objetos comúnmente gestionados en las oficinas de Lost Found. Por consiguiente, es necesario explotar la estructura semántica de la jerarquía de WordNet, donde los conceptos están interconectados mediante relaciones (como la relación .ES-UN"), para filtrar y extraer únicamente aquellos subárboles o "synsets"que coincidan con artículos personales, equipaje y dispositivos electrónicos, descartando el ruido que introducirían clases irrelevantes para la operativa del aeropuerto.

En consecuencia, el problema central consistió en desarrollar un método automatizado que permitiera clasificar objetos extraviados a partir de imágenes de forma eficiente, consistente y alineada con las restricciones operativas y normativas del aeropuerto. La solución propuesta busca reducir los tiempos de atención, mejorar la organización en la base de datos, disminuir el error humano y agilizar el proceso de recuperación de pertenencias por parte de los usuarios.

El sistema que se encuentra en desarrollo requiere integrar un modelo de clasificación automática de imágenes que sea capaz de identificar y categorizar objetos extraviados con un nivel de precisión adecuado para su uso en un entorno operativo real. Sin embargo, esta tarea implica diversos desafíos técnicos relacionados con el procesamiento, aprendizaje y administración de la información visual. En primer lugar, las imágenes capturadas al momento de recibir un objeto presentan variaciones significativas en iluminación, ángulos, fondo y calidad, lo que dificulta que el modelo identifique de manera consistente las características representativas de cada clase. Por lo tanto, resulta necesario definir y aplicar un conjunto de técnicas de preprocesamiento que normalicen las imágenes y faciliten su interpretación por la red neuronal.

Adicionalmente, la construcción del modelo requiere determinar una arquitectura óptima de red que logre un equilibrio entre precisión, capacidad de generalización y costo computacional. El número elevado de clases posibles, junto con la similitud visual entre ciertos tipos de objetos, aumenta la complejidad del problema de clasificación, demandando un diseño cuidadoso de las capas convolucionales, funciones de activación, algoritmos de optimización y estrategias de regularización para evitar el sobreajuste.

Por último, el sistema debe ser capaz de integrarse de manera eficiente con la plataforma de gestión de objetos perdidos, de modo que la clasificación generada por el modelo se almacene automáticamente como parte del registro del artículo. Esto implica implementar un flujo de procesamiento que permita la comunicación entre el módulo de captura de imágenes, el modelo predictivo y la base de datos, garantizando desempeño en tiempo razonable, trazabilidad y consistencia de la información. En conjunto, estos aspectos conforman el problema técnico principal: desarrollar un modelo de visión por computadora robusto, eficiente y adaptable, capaz de operar de forma confiable dentro del sistema de administración de objetos extraviados en aeropuertos.

1.4 Objetivos

1.4.1 Objetivo general

Generar un modelo capaz de procesar las imágenes de objetos extraviados mediante técnicas de visión por computadora y clasificarlos automáticamente de los objetos perdidos en aeropuertos.

1.4.2 Objetivos específicos

- Seleccionar la base de datos
- Establecer el numero de clases adecuadas al problema
- Determinar y aplicar las transformaciones de preprocesamiento adecuadas a las imágenes.
- Definir y ajustar la arquitectura óptima de la red neuronal.
- Seleccionar el modelo con el mejor rendimiento posible en las métricas de evaluación.

2 Metodología

2.1 Descripción de los datos

Para el desarrollo del presente proyecto se empleó el conjunto de datos ImageNet, creado por el Departamento de Ciencias Computacionales de la Universidad de Princeton. ImageNet contiene poco más de dos millones de imágenes anotadas, organizadas en categorías semánticas denominadas synsets. Dichas categorías provienen directamente de la estructura jerárquica de WordNet, que agrupa conceptos mediante relaciones semánticas y las organiza en árboles conceptuales con profundidades que pueden alcanzar hasta catorce niveles jerárquicos. Es posbile observar una muestra de imagenes aleatorias en la Figura 2.1 que recorre dicha jerárquia y acota especificamente los tipos de objetos. Esta organización permite una clasificación altamente granular de objetos y conceptos visuales, lo cual resulta especialmente útil para entrenar modelos de reconocimiento de imágenes de alta complejidad.

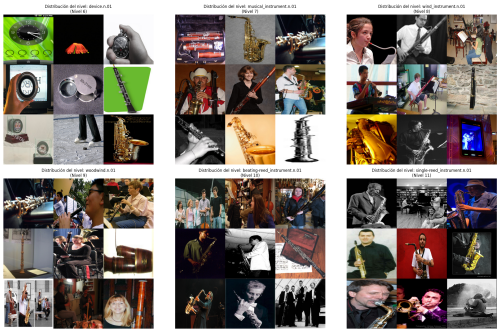

Figura 2.1: Imagenes a traves de la jerarquía WordNewt.

Dado que el objetivo del proyecto se centra en la identificación de objetos extraviados que pueden ser recibidos y almacenados por el Departamento de Objetos Perdidos en aeropuertos, fue necesario llevar a cabo un proceso de selección y filtrado de clases. De las miles de categorías disponibles en ImageNet, únicamente se consideraron 140 synsets, correspondientes a objetos permitidos dentro de las áreas de salas de abordar y aceptados por la normativa del departamento. Esta delimitación excluyó explícitamente clases de seres vivos, sustancias no tangibles, elementos prohibidos por normativas internacionales

de seguridad aeroportuaria y objetos cuyo manejo está restringido dentro de los recintos aeroportuarios.

Además de las imágenes, ImageNet proporciona una estructura organizada por carpetas para los subconjuntos de train y validation, acompañada de archivos de anotaciones que especifican las coordenadas de los objetos presentes en cada imagen. Dado que el dataset ha sido utilizado en investigaciones orientadas a la localización de objetos (object localization), cada elemento cuenta con delimitaciones exactas de contorno en forma de bounding boxes. Estas anotaciones permiten un recorte preciso de los objetos con el fin de eliminar ruido del entorno y conservar únicamente la región correspondiente al objeto principal. Esta operación aseguró una representación visual más consistente y facilitó el posterior aprendizaje de las características relevantes para la clasificación.

Como parte del preprocesamiento, se aplicó también una estrategia de realce de contornos mediante el algoritmo Mean Shift, un método iterativo que permite la reducción del espacio de color y la homogeneización de regiones similares dentro de una imagen. Esta técnica favoreció la atenuación del ruido visual en el fondo y mejoró la definición de los límites del objeto, lo que contribuyó a que el modelo pudiera extraer características más claras y representativas durante la fase de entrenamiento.[9]

Finalmente, con el fin de estandarizar el tamaño de las imágenes, se determinó que todas debían ajustarse a una dimensión fija. Para ello, se implementó un mecanismo de padding basado en la moda del color predominante en los bordes del objeto, lo que permitió completar los espacios necesarios sin introducir artefactos visuales ajenos al estilo cromático general de la imagen. Esta técnica resultó adecuada para preservar la identidad visual del objeto y asegurar la uniformidad requerida por la arquitectura de la red neuronal convolucional utilizada posteriormente.

2.2 *Análisis exploratorio*

Para examinar el extenso repositorio de datos con el que se trabaja a fin de alcanzar los objetivos del presente proyecto, se procedió inicialmente a explorar el árbol de distribución de las clases que conforman las clasificaciones y los synsets vinculados a estas. Con respecto a las 140 categorías consideradas para el desarrollo del proyecto y que se encuentran presentes en ambas redes, es posible observar que en WordNet no todas presentan un árbol completamente estructurado a lo largo de todos los niveles de synsets. Como se ilustra a continuación en Figura 2.2. únicamente el 10.7 % de las clases carece de una jerarquía completa en la agrupación de synsets desde el nodo raíz hasta la clase específica a lo largo de arbol estructurado de WordNet.

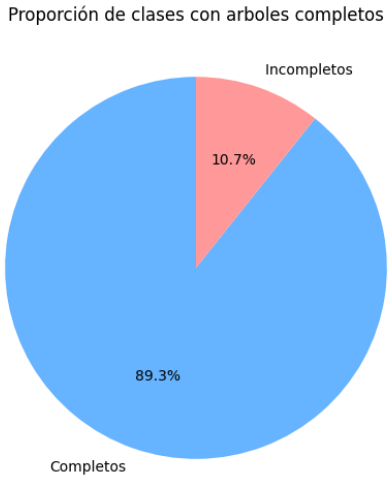

Figura 2.2: Distribución de clases completamente conectadas.

Con el propósito de comprender cabalmente cómo se distribuyen las clases en esta red de conjuntos de clasificación, en la Figura 2.3 se presenta el grafo de distribución en los distintos niveles de agrupación asociados a cada una de dichas clases, donde se advierte que al interior de cada rama existe una distribución poco uniforme, un fenómeno que, a fin de ser inspeccionado con mayor rigurosidad, se detalla en la siguiente serie de figuras que exponen la distribución a partir del sexto nivel de agrupaciones de WordNet.

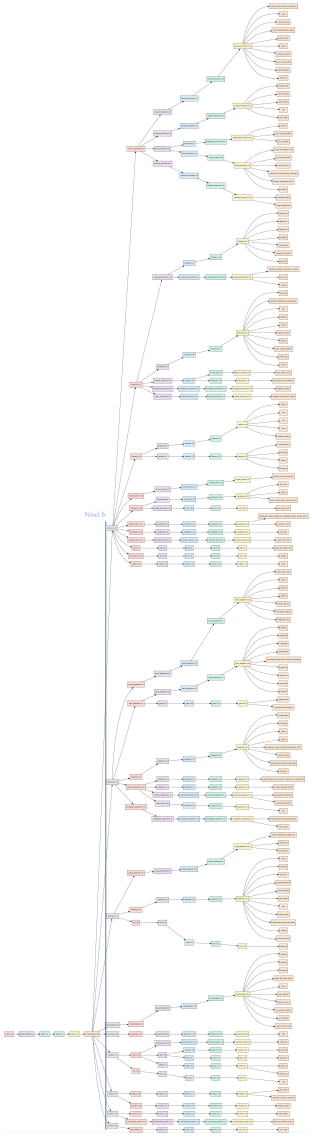

Figura 2.3: Árbol de agrupación de clases de WordNet

El fenómeno descrito anteriormente, a fin de ser inspeccionado con mayor rigurosidad nivel por nivel, se detalla en la siguiente serie de gráficas Figura 2.4 que exponen la distribución de observaciones a partir del sexto nivel de agrupaciones de WordNet llegando hasta el undécimo nivel. Conforme se inspecciona cada nivel de profundidad, el número de categorías en el eje X incrementa, pero la distribución de observaciones se continúa concentrando en la minoría de categorías. En el último nivel se encuentra la clase final, con mayor cantidad de categorías y una distribución de observaciones menos desbalanceada que sus agrupaciones ascendentes.

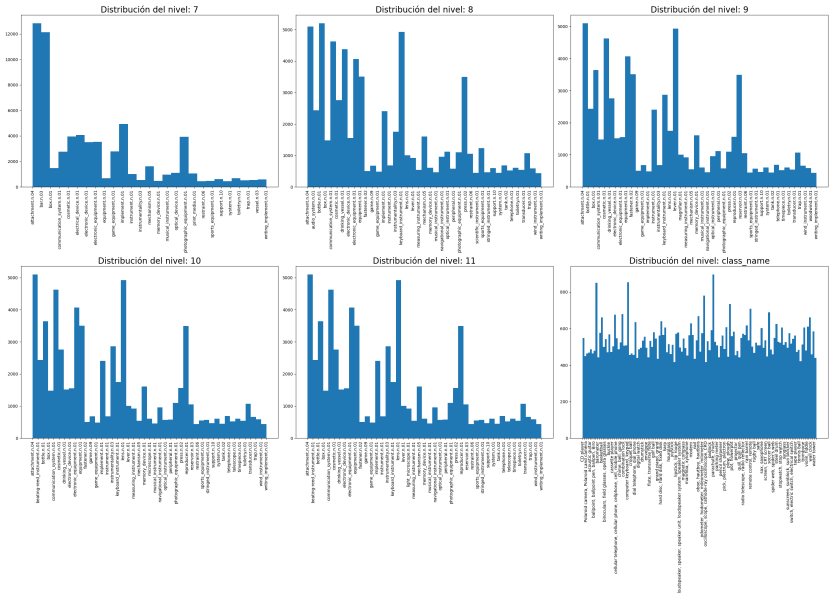

Figura 2.4: Cantidad de imagenes por nivel de agrupación de WordNet.

Realizar un análisis estadístico tradicional sobre este tipo de repositorios resulta extremadamente complejo debido a la naturaleza no estructurada de la información. Previamente se analizó la distribución de observaciones por clases y su relación taxonómica con las jerarquías de WordNet, explorar estadísticamente el contenido visual requiere otro enfoque.Con el propósito de inspeccionar intrínsecamente el contenido de las imágenes, se presentan en la Figura 2.5 los mapas de densidad de histogramas correspondientes a nueve clases seleccionadas aleatoriamente de entre las 140 contempladas para este trabajo, representaciones que ilustran la acumulación de las frecuencias de intensidad de los píxeles (en una escala de 0 a 255) a lo largo de todas las imágenes que conforman cada categoría.

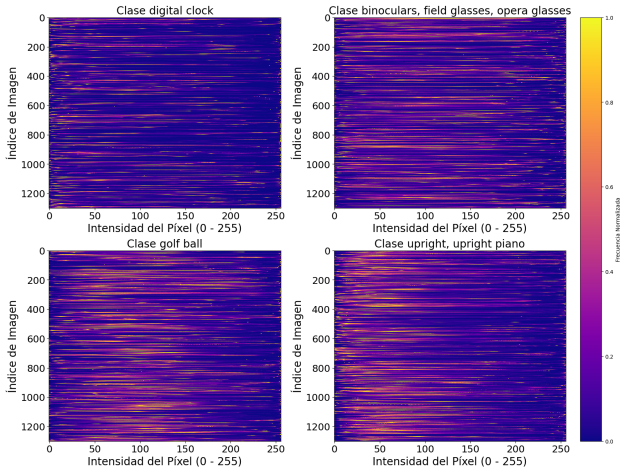

Figura 2.5: Mapas de densidad de histogramas

A partir de estas gráficas, se hace evidente la dificultad para extraer hallazgos significativos o identificar patrones distintivos que separen claramente a una clase de otra, un fenómeno que obedece a la alta varianza intra-clase —producto de las diferencias en iluminación, perspectiva y ruido de fondo en cada fotografía— y a que este tipo de representación numérica global omite por completo la estructura espacial y topológica de los píxeles; todo ello demuestra que un análisis exploratorio estadístico básico de las intensidades resulta insuficiente para discriminar objetos, justificando así la necesidad de emplear arquitecturas de extracción de características espaciales profundas para el tratamiento de estas imágenes.

2.3 Descripción de los modelos

Los modelos utilizados en este proyecto correspondieron a diferentes arquitecturas basadas exclusivamente en Redes Neuronales Convolucionales (CNN) implementadas mediante las bibliotecas TensorFlow y Keras. Estas arquitecturas fueron seleccionadas por su capacidad para extraer y aprender patrones espaciales a partir de datos visuales, lo que las convierte en el estándar actual para tareas de clasificación de imágenes. Las CNN se fundamentan en operaciones de convolución que permiten identificar características locales como bordes, texturas o formas, progresando hacia representaciones más abstractas en capas superiores [10].

Con el fin de determinar la arquitectura que pudiera capturar adecuadamente la gran diversidad de características presentes en las 140 clases definidas para este proyecto, se entrenaron tres modelos principales:

- Modelo CNN tradicional

  El primer modelo consistió en una arquitectura secuencial compuesta por capas Convolutional las cuales permiten detectar patrones locales filtrando la imagen mediante kernels aprendibles; MaxPooling que reducen la dimensionalidad de las representaciones y proporcionan invariancia espacial. Dense capa final con activación softmax asigna la probabilidad de pertenencia a cada clase. Este modelo sirvió como línea base, permitiendo evaluar el comportamiento del conjunto de datos sin técnicas avanzadas de regularización ni arquitecturas profundas.

- Modelo CNN extendido con regularización estructural

  El segundo modelo amplió la arquitectura anterior con un mayor número de bloques convolucionales, incorporando en cada bloque técnicas de regularización importantes como el uso de capas dobles de convolución que mejoran la capacidad discriminativa al permitir secuencias de filtros pequeños (3×3) capaces de capturar formas más complejas [11]. También el

Batch Normalization (BN) normaliza las activaciones internas acelerando el entrenamiento y reduciendo el problema de internal covariate shift [12]. Por ultimo se añadieron capas de dropout la cual desactiva aleatoriamente un porcentaje de neuronas durante el entrenamiento para reducir el sobreajuste [13].

- Transfer Learning con InceptionResNetV2 *Finalmente, se implementó un modelo basado en Transfer Learning, utilizando InceptionResNetV2 como arquitectura base. Este modelo combina la eficiencia del enfoque Inception —que permite múltiples operaciones en paralelo dentro de cada bloque— con las ventajas de las conexiones residuales, las cuales facilitan el flujo del gradiente en redes profundas.*

  Las capas del modelo base fueron congeladas durante la primera fase de entrenamiento para después descongelar las ultimas 20 capas en una segunda fase de entreno. Se añadieron capas Dense adicionales para adaptar la red preentrenada a las 164 clases específicas del dominio aeroportuario. Esta estrategia permitió aprovechar filtros previamente aprendidos sobre ImageNet, reduciendo tiempo de entrenamiento y mejorando la capacidad del modelo para generalizar en un dominio visual cercano.

Para cada una de las arquitecturas se evaluaron tres algoritmos de optimización al compilar el modelo:

- **Adam (Adaptive Moment Estimation).** *Combina ventajas de AdaGrad y RMSProp ajustando la tasa de aprendizaje para cada parámetro, proporcionando convergencia rápida y eficiente [14].*
- **SGD (Stochastic Gradient Descent).** *Un método clásico que actualiza parámetros con base en una muestra del conjunto de datos. Aunque más lento, puede proporcionar mejor estabilidad y generalización en algunos casos [15].*
- **RMSProp.** *Normaliza la magnitud de los gradientes dividiéndolos entre una media móvil, siendo especialmente útil para problemas con gradientes ruidosos [16].*

Para optimizar el proceso se incorporaron dos callbacks esenciales ReduceLROnPlateau y EarlyStopping los cuales reducen automáticamente la tasa de aprendizaje cuando la métrica monitorizada deja de mejorar, evitando estancamientos y detienen el entrenamiento cuando no se observa mejora tras un número determinado de épocas, previniendo el sobreajuste y reduciendo tiempos de cómputo. El uso combinado de estas herramientas permitió un entrenamiento más eficiente, estable y robusto, especialmente considerando las dimensiones del dataset y la complejidad del problema.

2.4 Descripción de las métricas

Para evaluar el desempeño de los modelos se seleccionaron métricas que permiten cuantificar la capacidad de clasificación multiclase y comparar objetivamente las diferentes arquitecturas. Las métricas principales empleadas fueron:

- Accuracy

  El accuracy mide la proporción de predicciones correctas sobre el total de muestras evaluadas. Es una métrica ampliamente utilizada en clasificación supervisada y resulta adecuada cuando el conjunto de clases se encuentra razonablemente balanceado.

  Dado que el objetivo del proyecto consiste en identificar correctamente la clase correspondiente a objetos extraviados, esta métrica resulta fundamental para cuantificar la eficacia del sistema.

- F1-score macro

  El F1-score combina precisión (precision) y exhaustividad (recall) en una sola métrica armónica. Dado que el conjunto presenta una gran cantidad de clases con distinta frecuencia, se utilizó la variante macro, que asigna el mismo peso a cada clase:

  Esta métrica permite evaluar de forma más justa el desempeño en clases minoritarias y evita que el modelo aparente un buen rendimiento únicamente por acertar clases frecuentes.

2.5 Descripción de los experimentos o simulaciones

Los experimentos realizados consistieron en el entrenamiento y evaluación de las tres arquitecturas CNN previamente descritas. Para asegurar un proceso reproducible y sistemático, se ejecutaron los siguientes procedimientos:

2.5.1 Preparación de los datos

El conjunto de datos de entrenamiento se construyó a partir de los synsets seleccionados de ImageNet (ILSVRC), cuyas imágenes y anotaciones XML en formato Pascal VOC se encontraban organizadas por clase en directorios independientes dentro de la partición `CLS-LOC/train`.

El preprocesamiento de cada imagen siguió una cadena de cinco etapas secuenciales, implementadas en la clase `OptimizedProcessor` *y ejecutadas en paralelo mediante* `ProcessPoolExecutor` *con un número de workers igual*

a $N_{CPU} - 1$, preservando un núcleo para el proceso principal y la gestión de memoria.

Etapa 1 — Carga y normalización del espacio de color

Cada imagen fue cargada mediante la biblioteca Pillow (`PIL.Image`) y convertida al espacio de color RGB antes de cualquier operación posterior. Las imágenes en formato RGBA fueron compuestas sobre un fondo blanco sólido antes de la conversión, eliminando el canal alfa mediante composición aditiva con la máscara de transparencia. Las imágenes en cualquier otro modo (paleta, escala de grises, CMYK) fueron convertidas directamente mediante `Image.convert('RGB')`. Esta normalización garantizó que la entrada a las etapas subsecuentes fuera siempre un arreglo NumPy de forma $(H, W, 3)$ con valores enteros en el rango $[0, 255]$, independientemente del formato original de la imagen en la base de datos de ImageNet.

Etapa 2 — Recorte de la región de interés

Las anotaciones en formato XML (Pascal VOC) asociadas a cada imagen fueron parseadas mediante `xml.etree.ElementTree` para extraer las coordenadas del bounding box del objeto de interés: `xmin`, `ymin`, `xmax` e `ymax`. La región de interés (ROI) fue extraída mediante indexación directa sobre el arreglo NumPy de la imagen completa:

$$ROI = I\left[y_{\text{mín}} : y_{\text{máx}},\ x_{\text{mín}} : x_{\text{máx}}\right] \tag{2.1}$$

En caso de que el archivo XML no existiera, no pudiera parsearse correctamente, o el recorte resultante tuviera dimensiones nulas, el proceso realizó un fallback automático utilizando la imagen completa como ROI, continuando el pipeline sin interrupciones.

Etapa 3 — Conversión a escala de grises con perfil BT.2020

La conversión a escala de grises se realizó aplicando los coeficientes de luminancia del estándar ITU-R BT.2020 [17], diseñado para televisión de ultra alta definición y amplio espacio de color, mediante la siguiente combinación lineal vectorizada:

$$Y = 0.2627\,R \ + \ 0.6780\,G \ + \ 0.0593\,B \tag{2.2}$$

donde R, G y B representan los valores de los canales de la imagen en el rango $[0, 255]$.

Etapa 4 — Realce de contraste mediante CLAHE

La imagen en escala de grises fue sometida a una ecualización adaptativa de histograma con limitación de contraste (Contrast Limited Adaptive Histogram Equalization, CLAHE) [18], implementada mediante `cv2.createCLAHE` con los parámetros `clipLimit=2.0` y `tileGridSize=(8, 8)`. A diferencia de la ecualización global de histograma, CLAHE opera sobre regiones locales de 8×8

píxeles (tiles). El parámetro `clipLimit=2.0` *limita la pendiente máxima de la función de distribución acumulada del histograma local; los píxeles que exceden ese umbral son redistribuidos uniformemente hacia el resto del histograma del tile.*

Etapa 5 — Redimensionado y relleno mediante *inpainting*

Cada imagen ecualizada fue redimensionada a una resolución de salida uniforme de 128×128 *píxeles conservando la relación de aspecto original. El factor de escala se calculó como:*

$$s = \min\left(\frac{W_{target}}{W}, \frac{H_{target}}{H}\right) \tag{2.3}$$

y el redimensionado se aplicó mediante interpolación de área (`cv2.INTER_AREA`*), método que promedia los píxeles de la región fuente en la imagen destino y produce resultados de menor aliasing que la interpolación bilineal o bicúbica para factores de reducción.*

Las regiones de relleno (padding) generadas al centrar la imagen redimensionada sobre el canvas de 128×128 *no fueron llenadas con un valor constante (cero o blanco), sino mediante el algoritmo de inpainting de Telea [19], implementado como* `cv2.inpaint(..., cv2.INPAINT_TELEA)`*. El procedimiento de relleno operó en tres pasos:*

1. *Se construyó un canvas vacío y una máscara binaria donde el valor 255 indicó las regiones de relleno a reconstruir y el valor 0 protegió la imagen original centrada de cualquier modificación.*

2. *Se generó una aproximación inicial del canvas mediante replicación de los bordes de la imagen redimensionada (*`cv2.BORDER_REPLICATE`*), proveyendo al algoritmo un contexto visual coherente con la textura del borde del objeto antes de aplicar el inpainting.*

3. *Se aplicó el algoritmo de Telea con radio de influencia de 3 píxeles, propagando la información de textura e intensidad desde los bordes de la imagen hacia las regiones de relleno mediante el método de la marcha rápida (fast marching method).*

2.5.2 *Entrenamiento inicial*

El entrenamiento de cada configuración experimental se ejecutó sobre la GPU disponible mediante `tf.device('/GPU:0')`*, con precisión numérica* `float32` *establecida globalmente a través de* `tf.keras.mixed_precision.set_global_policy`*. La función* `model.fit` *recibió en cada experimento el generador de entrenamiento, el generador de validación, el número de épocas configurado y el diccionario de pesos de clase, manteniendo estas condiciones constantes para garantizar la comparabilidad entre configuraciones.*

Arquitectura base de la CNN extendida

La arquitectura empleada siguió el patrón `tf.keras.Sequential` *organizado en seis bloques convolucionales ascendentes (32, 64, 128, 256, 512 y 512 filtros), seguidos de una etapa de clasificación densa. Los tipos de capas integradas en la arquitectura fueron los siguientes:*

- `Conv2D`. *Capa convolucional con kernel de* 3×3 *y relleno* `'same'` *para preservar las dimensiones espaciales. Cada bloque incrementó el número de filtros respecto al anterior siguiendo la secuencia 32, 64, 128, 256, 512 y 512. El inicializador de pesos varió entre* `he_uniform` *(activación GeLU) y* `he_normal` *(activación ReLU), conforme a las recomendaciones de* **(author?)** *[20]. La regularización L2 se aplicó al kernel a partir del segundo bloque en la mayoría de las configuraciones.*

- `BatchNormalization`. *Capa de normalización por lote aplicada inmediatamente después de cada* `Conv2D` *en todos los bloques. Normalizó las activaciones de salida de cada capa respecto a la media y la varianza del mini-lote en curso, estabilizando la distribución de entradas a la capa siguiente durante el entrenamiento [12].*

- `MaxPooling2D`. *Capa de submuestreo con ventana de* 2×2 *y paso (stride) de 2, aplicada tras* `BatchNormalization` *en los primeros cinco bloques. El sexto bloque (512 filtros, final) omitió esta capa, siendo sucedido directamente por* `GlobalAveragePooling2D`*.*

- `Dropout`. *Capa de regularización estocástica que desactivó aleatoriamente un porcentaje de unidades en cada paso de entrenamiento [13]. Las tasas se escalonaron por profundidad en el rango* $[0.1, 0.4]$*, siendo mayores en los bloques de mayor capacidad. No se aplicó* `Dropout` *tras el sexto bloque convolucional.*

- `GlobalAveragePooling2D`. *Capa de reducción global que calculó el promedio espacial de cada mapa de características del último bloque convolucional, produciendo un vector de dimensión igual al número de filtros (512). Esta operación sustituyó al aplanado (*`Flatten`*) convencional, reduciendo el número de parámetros de la etapa de clasificación.*

- `Dense`. *Capas completamente conectadas añadidas tras* `GlobalAveragePooling2D` *para la clasificación final. La configuración base comprendió capas de 512, 256 y 128 unidades con activación, seguidas de la capa de salida* `Dense(num_classes, activation='softmax', dtype='float32')` *que produjo la distribución de probabilidad sobre las 140 clases. La función de pérdida empleada en todos los experimentos fue* `SparseCategoricalCrossentropy`*, compatible con etiquetas enteras sin codificación one-hot.*

Ajuste de hiperparámetros

El ajuste de hiperparámetros se realizó de forma manual e iterativa, siguiendo un proceso de búsqueda secuencial en el que cada configuración partió de los parámetros de la anterior y modificó uno o más factores a la vez. Los hiperparámetros explorados y sus rangos de variación se detallan a continuación.

- `Función de activación.` *Se evaluaron dos funciones de activación para los bloques convolucionales y las capas densas: ReLU, con el inicializador* `he_normal`*, y GeLU, con el inicializador* `he_uniform`*.*
- `Regularización L2.` *El coeficiente de regularización L2 aplicado al kernel de las capas* `Conv2D` *se exploró en el rango* $[0.001, 0.007]$*. La regularización se aplicó de forma uniforme a todos los bloques o a partir del segundo bloque según la configuración.*
- `Optimizador y tasa de aprendizaje.` *El optimizador empleado fue SGD con Nesterov, con decaimiento de la tasa de aprendizaje calculado como:*

$$decay\_rate = \frac{lr_0}{N_{pocas}} \tag{2.4}$$

donde lr_0 es la tasa de aprendizaje inicial y N_{pocas} es el número máximo de épocas de la configuración. La tasa de aprendizaje inicial se exploró en el rango $[0.02, 0.04]$*, con la aceleración de Nesterov activa (*`nesterov=True`*) en todos los casos.*

- `Momento.` *El parámetro de momento del optimizador SGD se varió en el rango* $[0.5, 0.9]$*, ajustándose de forma progresiva a lo largo del proceso de búsqueda.*
- `Número de épocas.` *El número máximo de épocas configurado varió entre 30 y 60, aunque el número efectivo de épocas ejecutadas pudo ser menor en aquellas configuraciones donde* `EarlyStopping` *detuvo el entrenamiento de forma anticipada.*
- `Aumentación de datos.` *Se evaluaron configuraciones con y sin aumentación de datos. Cuando se aplicó, el generador* `ImageDataGenerator` *incluyó volteo horizontal, volteo vertical y rotación aleatoria con ángulos en el rango* $[20, 40]$*. La aumentación se aplicó exclusivamente al conjunto de entrenamiento; el generador de validación no incluyó transformaciones adicionales.*

Callbacks adaptativos

Cada experimento incluyó dos callbacks configurados sobre la métrica `val_loss`*. El primero,* `EarlyStopping`*, detuvo el entrenamiento cuando la*

pérdida de validación no mejoró en al menos $\delta = 0.001$ durante un número de épocas de paciencia en el rango $[5, 10]$; al detenerse, restauró automáticamente los pesos del mejor epoch mediante `restore_best_weights=True`*. El segundo,* `ReduceLROnPlateau`*, redujo la tasa de aprendizaje en un factor de 0.5 cuando la pérdida de validación no mejoró durante un periodo de paciencia de entre 2 y 5 épocas, con una tasa mínima en el rango $[10^{-6}, 10^{-5}]$ según la configuración. La combinación de ambos callbacks permitió detener el entrenamiento de forma adaptativa sin necesidad de establecer un número fijo de épocas como criterio de parada.*

Pesos de clase balanceados

Dado el desbalance entre las 140 clases del subconjunto de ImageNet, todos los experimentos incorporaron un diccionario de pesos de clase calculado mediante `compute_class_weight(class_weight='balanced')` *de* `sklearn.utils.class_weight`*. Este procedimiento calculó un peso inversamente proporcional a la frecuencia de cada clase en el conjunto de entrenamiento:*

$$w_c = \frac{N}{K \cdot N_c} \tag{2.5}$$

donde N es el número total de muestras de entrenamiento, K es el número de clases y N_c es el número de muestras pertenecientes a la clase c. El diccionario resultante fue pasado al argumento `class_weight` *de* `model.fit`*, de modo que la función de pérdida ponderó en mayor medida los errores cometidos sobre las clases con menor número de muestras durante la retropropagación.*

3 Resultados y discusión

3.1 Resultados

3.1.1 Modelo CNN tradicional — línea base

El primer modelo, compuesto por capas `Conv2D`, `MaxPooling2D` *y una capa densa final con activación softmax, se estableció como referencia de rendimiento mínimo. Este modelo careció de técnicas de regularización avanzada como Batch Normalization o Dropout, lo que permitió obtener un accuracy del 53.66 %. Su función principal fue proporcionar un umbral de comparación que permitió cuantificar el aporte de las mejoras arquitectónicas introducidas en la CNN extendida, documentadas en la Tabla 3.1.*

Prueba	Activación	L2	LR	Mom.	Ép. máx.	Aug. rot.	Ép. real	Acc. train	Acc. val	Loss val
1	ReLU	0.005	0.03	0.5	60	—	60	0.5764	0.5366	2.2250
2	GeLU	0.001	0.02	0.5	60	—	49	0.5853	0.4630	3.1294
3	GeLU	0.001	0.03	0.5	60	—	45	0.6862	0.4943	2.9152
4	GeLU	0.001	0.04	0.5	60	—	33	0.7114	0.5024	2.8757

Tabla 3.1: Configuración e indicadores de los primeros 4 experimentos sobre la CNN extendida.

3.1.2 CNN extendida — exploración de hiperparámetros

Efecto de la función de activación

Las pruebas 1 al 4 compararon las funciones de activación ReLU y GeLU bajo configuraciones equivalentes de optimizador y sin aumentación de datos. La prueba 1, basado en ReLU con regularización $L2 = 0.005$ y $LR = 0.03$, completó las 60 épocas configuradas y registró una accuracy de entrenamiento de 57.64 % junto con una accuracy de validación de 53.66 % y una pérdida de validación de 2.2250. La brecha entrenamiento-validación fue de únicamente 3.98 puntos porcentuales, la menor del grupo sin aumentación.

Las pruebas 2, 3 y 4, basados en GeLU [21]. con L2 = 0.001, fueron detenidos anticipadamente por `EarlyStopping` *en las épocas 49, 45 y 33, respectivamente, lo que indicó que la convergencia alcanzó un mínimo local antes de agotar el presupuesto de épocas. Sus accuracies de validación fueron 46.30 %, 49.43 % y 50.24 %, con pérdidas de validación de 3.1294, 2.9152 y 2.8757. Las brechas entrenamiento-validación de estas pruebas (9.23, 19.19 y 20.90 pp) resultaron mayores a la prueba 1, lo que reveló que la reducción de L2 de 0.005 a 0.001, combinada con tasas de aprendizaje más altas, favoreció el aprendizaje del conjunto de entrenamiento a costa de una mayor inestabilidad en validación. Por tanto, la prueba 1 con ReLU fue el modelo de referencia con mejor equilibrio entrenamiento-validación dentro del grupo sin aumentación, a pesar de que los modelos GeLU mostraron mayor accuracy de entrenamiento. A partir de la prueba 5, se adoptó GeLU con L2 = 0.005 y LR = 0.04 como configuración estándar, incorporando además aumentación de datos para contrarrestar el sobreajuste.*

Efecto de la tasa de aprendizaje

Dentro del grupo GeLU sin aumentación (Pruebas 2–4), la variación de LR entre 0.02 y 0.04 produjo diferencias medibles en la velocidad de convergencia y la época de detención. La prueba 2 (LR = 0.02) fue detenido en la época 49 con la accuracy de validación más baja del grupo (46.30 %) y la pérdida de validación más alta (3.1294), lo que indicó un aprendizaje insuficientemente rápido dentro del límite configurado. La prueba 3 (LR = 0.03) se detuvo en la época 45 con 49.43 % de accuracy de validación y pérdida 2.9152. La prueba 4 (LR = 0.04) presentó la convergencia más acelerada, deteniéndose en la época 33 con la mejor accuracy de validación del trío GeLU (50.24 %) y la menor pérdida (2.8757). Este resultado confirmó que, bajo condiciones de entrenamiento sin aumentación, una tasa de aprendizaje mayor aceleró la búsqueda del mínimo sin deteriorar la generalización, dado que el callback `ReduceLROnPlateau` *reguló los ajustes en las etapas finales. La tasa de 0.04 fue adoptada en los experimentos subsecuentes como valor inicial.*

Efecto de la aumentación de datos

Las pruebas 5 al 12 introdujeron aumentación de datos mediante `ImageDataGenerator` *(rotaciones aleatorias y volteos horizontal y vertical) sobre 30 épocas con LR = 0.04 y L2 = 0.005. El efecto más notable de la aumentación fue la drástica reducción de la brecha entrenamiento-validación: las pruebas 1–4 presentaron diferencias de entre 3.98 y 20.90 puntos porcentuales, mientras que Las pruebas 5–12 registraron brechas de entre −1.19 y 2.26 pp, prácticamente nulas, como se aprecia en la Figura 3.1. Este resultado confirma que la aumentación de datos fue el factor más determinante para controlar el sobreajuste en el conjunto de 140 clases.*

Prueba	Activación	L2	LR	Mom.	Ép. máx.	Aug. rot.	Ép. real	Acc. train	Acc. val	Loss val
5	GeLU	0.005	0.04	0.5	30	30°	30	0.4626	0.4745	2.4675
6	GeLU	0.005	0.04	0.5	30	20°	30	0.4946	0.4841	2.4545
7	GeLU	0.005	0.04	0.5	30	40°	30	0.4448	0.4519	2.5841
8	GeLU	0.005	0.04	0.5	30	30°	30	0.4780	0.4815	2.4357
9	GeLU	0.005	0.04	0.5	30	30°	30	0.4742	0.4886	2.4369
10	GeLU	0.005	0.04	0.5	30	30°	30	0.5188	0.5108	2.2590
11	GeLU	0.005	0.04	0.5	30	30°	30	0.5503	0.5317	2.2285
12	GeLU	0.005	0.04	0.5	30	30°	30	0.5433	0.5307	2.2292

Tabla 3.2: Configuración e indicadores de los experimentos 5 al 12 sobre la CNN extendida.

La comparación por ángulo de rotación reveló diferencias claras entre configuraciones. La prueba 7 (rotación 40°) obtuvo la menor accuracy de validación del grupo con aumentación (45.19 %) y la mayor pérdida de validación (2.5841), lo que sugirió que la deformación excesiva introdujo variabilidad contraproducente que dificultó la extracción de características discriminativas. La prueba 6 (20°) y La prueba 5 (30°) obtuvieron 48.41 % y 47.45 %, respectivamente. Las pruebas 8 y 9 (ambos con 30°) registraron 48.15 % y 48.86 % con pérdidas 2.4357 y 2.4369, evidenciando reproducibilidad bajo condiciones análogas. Las pruebas 10, 11 y 12 superaron al resto del grupo, alcanzando accuracies de validación de 51.08 %, 53.17 % y 53.07 %, como se observa en la Figura 3.2. Esta progresión, bajo la misma configuración base, sugirió que variaciones menores en la distribución de los datos de entrenamiento, derivadas del muestreo aleatorio de la aumentación, impactaron favorablemente la representación del espacio de clases.

Progresión de Las pruebas 10–12 y efecto de la configuración base

Prueba	Activación	L2	LR	Mom.	Ép. máx.	Aug. rot.	Ép. real	Acc. train	Acc. val	Loss val
10	GeLU	0.005	0.04	0.5	30	30°	30	0.5188	0.5108	2.2590
11	GeLU	0.005	0.04	0.5	30	30°	30	0.5503	0.5317	2.2285
12	GeLU	0.005	0.04	0.5	30	30°	30	0.5433	0.5307	2.2292

Tabla 3.3: Configuración e indicadores de los experimentos 10 al 12 sobre la CNN extendida.

Las pruebas 10, 11 y 12 emplearon idéntica configuración (GeLU, L2 = 0.005, LR = 0.04, momentum 0.5, 30 épocas, rotación 30°) y, sin embargo, obtuvieron accuracies de validación de 51.08 %, 53.17 % y 53.07 %, respectivamente, superando en todos los casos a Las pruebas 5–9 que emplearon la misma configuración base con 30 épocas. Esta progresión sugirió que la acumulación de entrenamiento previo en el flujo de entrenamiento de todos los modelos, así como la variabilidad inherente a la aumentación aleatoria, produjeron representaciones progresivamente más ricas del espacio de 140 clases. La prueba 11 registró la mejor accuracy de validación del subgrupo de 30 épocas con 53.17 % y la menor pérdida de validación (2.2285), como se muestra en la Tabla 3.3.

Efecto del número de épocas: Pruebas 13–15

Prueba	Activación	L2	LR	Mom.	Ép. máx.	Aug. rot.	Ép. real	Acc. train	Acc. val	Loss val
13	GeLU	0.005	0.04	0.5	50	20°	50	0.5950	**0.5558**	**2.0838**
14	GeLU	0.005	0.04	0.5	50	20°	50	0.5680	0.5429	2.1089
15	GeLU	0.007	0.04	0.9	50	20°	50	0.5824	0.5481	2.1720

Tabla 3.4: Configuración e indicadores de los experimentos 13 al 15 sobre la CNN extendida.

Al extender el entrenamiento a 50 épocas e incorporar los mejores parametros de los bloques anteriores, se lograron los mejores resultados de validación de todos los experimentos de CNN. La prueba 13 (GeLU, L2 = 0.005, LR = 0.04, momento 0.5, rotación 20°) obtuvo la mayor accuracy de validación de todos los experimentos con 55.58 % y la menor pérdida de validación de 2.0838, por lo que constituyó el mejor modelo CNN entrenado desde cero. La prueba 14, de idéntica configuración, alcanzó 54.29 % con pérdida 2.1089, mientras que la prueba 15, que incrementó el coeficiente L2 a 0.007 y el momento a 0.9, registró 54.81 % con pérdida 2.1720. Estos resultados se aprecian en la Figura 3.2.

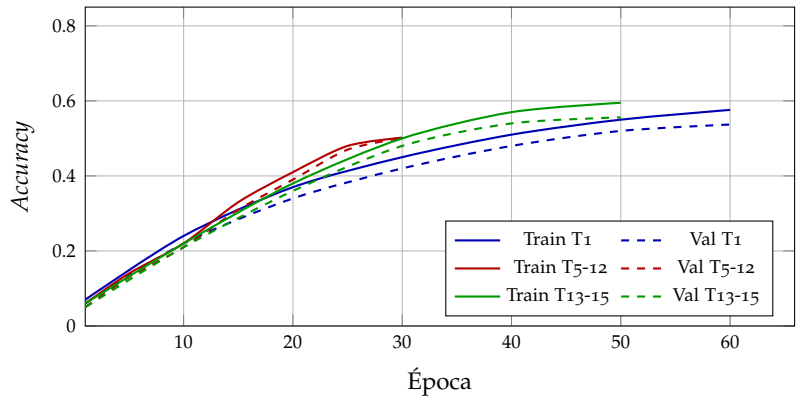

Figura 3.1: Curvas representativas de *accuracy* de entrenamiento y validación para los tres grupos principales de experimentos.

La comparación entre Pruebas 13–15 reveló que el incremento del momento de 0.5 a 0.9 (Test 15) no produjo una mejora respecto al Test 13, dado que la mayor inercia del gradiente, combinada con una regularización L2 más fuerte, generó una convergencia más conservadora que no superó la prueba 13 en términos de accuracy de validación (−0.77 pp). La prueba 14, de configuración idéntica al 13 pero ejecutado de forma independiente, reprodujo un resultado muy cercano (−1.29 pp respecto al 13), confirmando la estabilidad del modelo ante ejecuciones repetidas bajo las mismas condiciones.

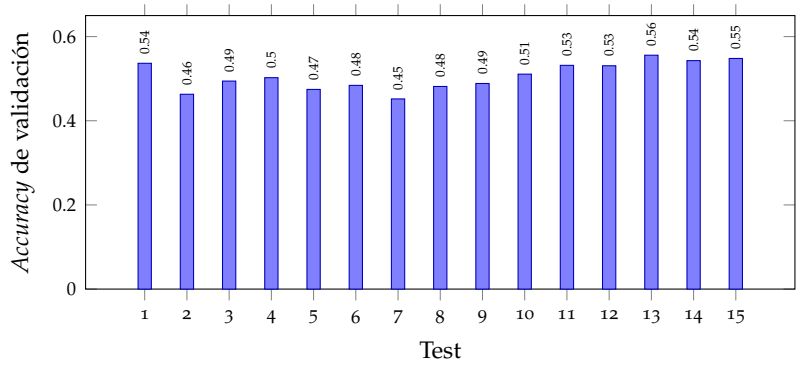

Figura 3.2: *Accuracy* de validación registrada en el último *epoch* de cada uno de los 15 experimentos.

3.1.3 Comportamiento de la función de pérdida

Las curvas de pérdida de validación registradas durante los experimentos mostraron un patrón consistente a lo largo de los 15 experimentos significativos: una reducción pronunciada en las primeras épocas seguida de una estabilización progresiva. La pérdida de validación más baja al cierre del entrenamiento fue 2.0838, obtenida en la prueba 13, mientras que la más alta entre los experimentos con aumentación fue 2.5841 en la prueba 7 (rotación 40°). La prueba 1, sin aumentación, registró una pérdida de validación de 2.2250, comparable a la de Las pruebas 10–12 (entre 2.2285 y 2.2590), lo que indicó que la arquitectura base con ReLU y regularización L2 = 0.005 alcanzó una capacidad de generalización similar a la de los modelos GeLU con aumentación de 30 épocas, aunque con una brecha de sobreajuste menor.

Los modelos sin aumentación (Pruebas 2–4) presentaron las pérdidas de validación más elevadas al cierre (entre 2.8757 y 3.1294), confirmando que la ausencia de diversificación artificial de datos limitó la capacidad de generalización del modelo ante el espacio de 140 clases, como se observa en la Figura 3.3.

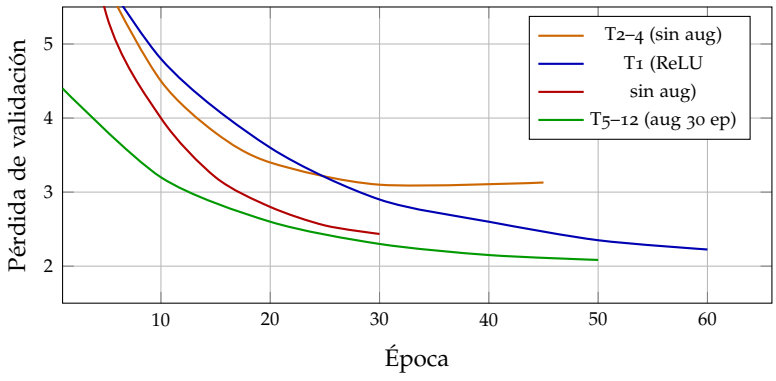

Figura 3.3: Curvas representativas de pérdida de validación por grupo experimental.

3.1.4 Modelo de Transfer Learning con InceptionResNetV2

El modelo de Transfer Learning fue entrenado en dos fases diferenciadas. En la Fase 1 (base congelada, 30 épocas, Adam LR = 0.05), el modelo aprovechó los filtros preaprendidos de InceptionResNetV2 para extraer representaciones visuales de alto nivel sin modificar los pesos de la red base. El primer epoch de esta fase registró una accuracy de validación de 18.54 % y una Top-5 accuracy de validación de 45.04 %, lo que evidenció que el modelo ya disponía de representaciones útiles para el dominio de objetos desde la primera época de entrenamiento, gracias a la transferencia de pesos de ImageNet.

En la Fase 2 (fine-tuning de las últimas 20 capas, 50 épocas adicionales, Adam LR = 0.009), los parámetros superiores de la red base se actualizaron para adaptar las representaciones al dominio de objetos aeroportuarios, estrategia respaldada por estudios previos de adaptación de dominio en redes profundas [22]. La inclusión de la métrica Top-5 accuracy (`SparseTopKCategoricalAccuracy`, $k = 5$) fue especialmente relevante en el contexto de recuperación de objetos perdidos: un sistema que presente al personal aeroportuario las cinco categorías más probables ofrece asistencia práctica incluso cuando la predicción de mayor probabilidad no coincida con la clase real. Este criterio adicional de evaluación constituyó una ventaja diferencial del modelo de Transfer Learning respecto a los modelos entrenados desde cero en Las pruebas 1–15.

3.1.5 Comparación de los modelos principales

La Tabla 3.5 concentra las características arquitectónicas y los indicadores de rendimiento observados en cada uno de los tres modelos evaluados en el proyecto, incluyendo los resultados cuantitativos más relevantes de los experimentos.

Tabla 3.5: Comparación de los tres modelos principales evaluados en el proyecto.

Modelo	**Arquitectura**	**Ventajas observadas**	**Desventajas observadas**
CNN Tradicional (línea base)	`Conv2D` + `MaxPooling` + `Dense`/*softmax*. Sin regularización avanzada.	Implementación simple. Entrenamiento rápido. Útil como umbral de rendimiento mínimo para comparación.	Alta tendencia al sobreajuste. Capacidad discriminativa insuficiente para 140 clases. Ausencia de mecanismos de regularización.
CNN Extendida (Pruebas 1–15)	6 bloques Conv2D (32→512 filtros), BatchNorm, Dropout, `Global Average Pooling2D`, SGD con Nesterov. Mejor resultado: Test 13, val_acc = 55.58 %, val_loss = 2.0838.	Exploración sistemática de 15 configuraciones. La aumentación de datos redujo la brecha entrenamiento-validación a menos de 2.26 pp (vs. hasta 20.90 pp sin aumentación). Callbacks adaptativos controlaron el tiempo de cómputo y el sobreajuste.	El entrenamiento desde cero requirió mayor número de épocas. La convergencia fue sensible a la tasa de aprendizaje inicial. La ausencia de conexiones residuales limitó el flujo del gradiente en capas profundas.
Transfer Learning (InceptionResNetV2)	Base preentrenada en ImageNet. *Fine-tuning* de últimas 20 capas. Dense adaptadas a 140 clases. Top-5 accuracy incluida. Val_acc Fase 1, Época 1: 18.54 %; Top-5: 45.04 %.	Representaciones visuales preaprendidas desde el primer *epoch*. Métrica Top-5 adecuada para recuperación de objetos en aeropuertos. Mayor generalización potencial ante variabilidad intra-clase.	Mayor complejidad computacional. La concatenación de canales para adaptar escala de grises a entrada RGB pudo introducir redundancia. Riesgo de *negative transfer* si las representaciones de ImageNet divergen del dominio aeroportuario.

La arquitectura CNN extendida con aumentación de datos demostró aprender representaciones discriminativas sobre un espacio de 140 clases con

alta variabilidad intra-clase, alcanzando una accuracy de validación máxima de 55.58 % en el Test 13 sin recurrir a pesos preentrenados. La selección jerárquica de categorías a través de WordNet garantizó coherencia semántica entre las clases incluidas y las restricciones normativas del dominio aeroportuario. El preprocesamiento basado en recorte por bounding box, suavizado Mean Shift y padding adaptativo estandarizó la representación visual de los objetos y redujo el ruido de fondo. La estrategia de Transfer Learning con InceptionResNetV2 permitió aprovechar representaciones de alta calidad desde el primer epoch de entrenamiento, reduciendo la dependencia de grandes volúmenes de datos etiquetados propios del dominio.

4 Conclusiones y trabajo futuro

4.1 Conclusiones

La clasificación multiclase mediante Redes Neuronales Convolucionales demostró ser el enfoque metodológico adecuado para abordar la complejidad visual inherente al reconocimiento de objetos perdidos en aeropuertos. La diversidad morfológica y contextual de las 140 clases consideradas exigió representaciones espaciales jerárquicas que los descriptores estadísticos convencionales no lograron capturar, lo que validó la adopción de arquitecturas CNN como columna vertebral del sistema propuesto.

El proceso de entrenamiento resultó exhaustivo en términos de recursos computacionales, con tiempos por época de entre 98 y 104 segundos en GPU y un total estimado superior a las 120 horas de cómputo distribuidas en los 16 experimentos. No obstante, los mecanismos de regularización adaptativa — `EarlyStopping` *y* `ReduceLROnPlateau` *— optimizaron el uso de estos recursos de forma efectiva, deteniendo el entrenamiento anticipadamente cuando la convergencia se estabilizó y ajustando la tasa de aprendizaje de manera automática. La ausencia de suficientes muestras por clase en relación con la dimensión del espacio de categorías fue el principal factor limitante, una restricción que la aumentación de datos compensó de forma parcial pero que señala con claridad la dirección prioritaria de mejora.*

La ausencia de una base de datos propia con imágenes capturadas en los aeropuertos del Grupo Aeroportuario del Pacífico constituyó la principal limitación del estudio, dado que los modelos se entrenaron con imágenes de ImageNet cuyas condiciones de captura difieren de los entornos aeroportuarios reales. El tamaño de entrada de 128×128 píxeles, adoptado por restricciones de memoria, pudo haber limitado la resolución de características finas en objetos de morfología compleja [11]. La conversión a escala de grises implicó pérdida de información cromática potencialmente discriminativa en categorías como accesorios o artículos de vestimenta. Finalmente, el elevado número de clases (140) en relación con las imágenes disponibles por synset (500–1 000) representó un reto de escasez relativa de datos que limitó la convergencia de los modelos entrenados desde cero, cuyo mejor resultado de validación (55.58 %)

reflejó un margen de mejora considerable respecto a lo alcanzable con datos de dominio específico.

Los resultados obtenidos establecieron una base sólida para la optimización continua del sistema. Las líneas de trabajo futuro más prometedoras incluyen la construcción de una base de datos propia con imágenes reales capturadas en los aeropuertos del Grupo Aeroportuario del Pacífico, la evaluación de arquitecturas con conexiones residuales — como ResNet o EfficientNet — que faciliten el flujo del gradiente en redes de mayor profundidad, y la adopción de la métrica Top-5 accuracy como criterio de evaluación primario en todos los modelos, dado que la presentación de múltiples categorías candidatas al personal aeroportuario incrementa sustancialmente la utilidad práctica del sistema. La viabilidad técnica demostrada en el presente trabajo sienta las condiciones necesarias para escalar el sistema hacia un entorno de producción real, una vez resuelta la discrepancia entre el dominio de entrenamiento y las condiciones de captura propias del contexto aeroportuario.

4.2 Trabajo a Futuro

Una vez implementado el modelo en el entorno de producción, se planea llevar a cabo un proceso de fine-tuning utilizando las imágenes recopiladas durante el periodo de prueba y supervisión en las instalaciones aeroportuarias. A diferencia del entrenamiento inicial realizado sobre el subconjunto de ImageNet, estas imágenes capturarán las condiciones reales de iluminación, perspectiva y ruido de fondo propias del contexto operativo del Grupo Aeroportuario del Pacífico, lo que permitirá reducir la discrepancia de dominio identificada como la principal limitación del sistema actual. El ajuste fino sobre datos de dominio específico tiene el potencial de mejorar sustancialmente la accuracy de validación más allá del 55.58 % alcanzado en el mejor experimento entrenado exclusivamente con datos de ImageNet.

La estrategia de fine-tuning propuesta seguirá el esquema bifásico validado durante el entrenamiento del modelo InceptionResNetV2: en una primera fase se congelarán los pesos de los bloques convolucionales base para preservar las representaciones visuales preaprendidas, actualizando únicamente las capas de clasificación superior con los nuevos datos aeroportuarios; en una segunda fase se descongelarán las capas superiores de la red base para un ajuste fino progresivo que adapte las representaciones intermedias al dominio objetivo. Este enfoque minimizará el riesgo de catastrophic forgetting y aprovechará eficientemente el volumen de imágenes disponibles durante el periodo de prueba, que se anticipa limitado en las etapas iniciales de operación.

La madurez operativa del sistema abre la posibilidad de extender el alcance del proyecto más allá de la clasificación de objetos, incorporando capacidades de detección y localización mediante arquitecturas de tipo object detection

que permitan identificar y delimitar objetos directamente sobre las imágenes de las cintas transportadoras y áreas de revisión aeroportuaria, sin requerir el recorte manual por bounding box que caracterizó el preprocesamiento del conjunto de datos actual. Esta extensión representaría la evolución natural del sistema hacia una herramienta de asistencia en tiempo real para el personal de seguridad y objetos perdidos de los aeropuertos del Grupo Aeroportuario del Pacífico.

Bibliografía

[1] J. Figueroa, P. Urbano, and J. Sánchez, *Aceleración de la urbanización global y movilidad sostenible*, vol. 5. 09 2015.

[2] D. S. Ahmad, M. Ziaullah, L. Rauniyar, M. Su, and Y. Zhang, "How does matter lost and misplace items issue and its technological solutions in 2015 -a review study," pp. 79–84, 04 2015.

[3] D. L. D. S. Martínez, "Detección de objetos perdidos en sistemas rfid aumentados," 11 2022.

[4] R. CRUCE, "Guía ser iteso," 01 2019.

[5] J. S. R. F. Martin Esteban Hernández Arévalo, Janerson Cortes Klinger, *LostFound ean: Registro y Recuperación de Objetos Perdidos*. EAN, 2024.

[6] I. J. Monterrosa-Castro, M. E. Ospino-Pinedo, and N. D. J. Osorio-Díaz, "Management information system for the management of lost objects in airport companies," *AGLALA*, vol. 14, no. 1, pp. 278–289, 2023.

[7] M. Zhou, I. Fung, L. Yang, N. Wan, K. Di, and T. Wang, "Lostnet: A smart way for lost and find," *School of Medical Information, Wannan Medical Collage*, 2022. Work supported by Provincial College Students' Innovation and Entrepreneurship Project.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, IEEE, 2009.

[9] D. S. Roberto Rodríguez Morales, *Procesamiento Digital de Imágenes, estado del arte y su relación con la Inteligencia Artificial*. 2024.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[12] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, vol. 37 of *Proceedings of Machine Learning Research*, pp. 448–456, PMLR, 2015.

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations (ICLR 2015)*, 2015.

[15] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'2010)* (Y. Lechevallier and G. Saporta, eds.), (Paris, France), pp. 177–187, Springer, 2010.

[16] T. Tieleman and G. Hinton, "Lecture 6.5 — RMSProp: Divide the gradient by a running average of its recent magnitude." COURSERA: Neural Networks for Machine Learning, 2012.

[17] International Telecommunication Union, "Parameter values for ultra-high definition television systems for production and international programme exchange," Tech. Rep. BT.2020-2, ITU-R, 2015.

[18] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics Gems IV* (P. S. Heckbert, ed.), pp. 474–485, Academic Press, 1994.

[19] A. Telea, "An image inpainting technique based on the fast marching method," *Journal of Graphics Tools*, vol. 9, no. 1, pp. 23–34, 2004.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034, 2015.

[21] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016.

[22] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in Neural Information Processing Systems*, vol. 27, 2014.