

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE SISTEMAS

UNIDAD DE TITULACIÓN

**RECONOCIMIENTO DEL CLIMA EN IMÁGENES UTILIZANDO
APRENDIZAJE PROFUNDO AUTOSUPERVISADO**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL GRADO DE
MAGISTER DE INVESTIGACIÓN EN COMPUTACIÓN**

DIEGO ANDRÉS ACUÑA ESCOBAR

diego.acuna@epn.edu.ec

Director: MSc. MARIA MONSERRATE INTRIAGO PAZMIÑO

monserrate.intriago@epn.edu.ec

2023

APROBACIÓN DEL DIRECTOR

Como director del trabajo de titulación RECONOCIMIENTO DEL CLIMA EN IMÁGENES UTILIZANDO APRENDIZAJE PROFUNDO AUTOSUPERVISADO, desarrollado por Diego Andrés Acuña Escobar, estudiante del programa de maestría de investigación en computación, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, doy por aprobada la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la Defensa oral.

MSc. MONSERRATE INTRIAGO PAZMIÑO

DIRECTOR

DECLARACIÓN DE AUTORÍA

Yo, DIEGO ANDRÉS ACUÑA ESCOBAR, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

DIEGO ANDRÉS ACUÑA ESCOBAR

DEDICATORIA

Este trabajo lo dedico especialmente a mis padres, Medardo y Rosita; y abuelitos, quienes con todo su amor y comprensión lograron enseñarme que con esfuerzo y perseverancia todo es posible cuando se tiene un objetivo claro. A mis hermanas, cuñado y sobrinos Dany y Maty. A mi compañera de vida Navila y a nuestros dos hermosos hijos Dannita Jey y Dieguito Jr. A mis primos y tíos que siempre estuvieron tendiéndome su mano cuando más lo necesité. Y finalmente, a mi suegra y cuñados quienes enseñan que en la vida siempre hay momentos para sonreír y a mi suegro en el cielo.

AGRADECIMIENTO

Con infinita gratitud y amor, expreso mi mayor sentimiento de agradecimiento a todos quienes aportaron en la consecución de esta nueva meta. Gracias a todos por confiar en mí y brindarme su apoyo en cada paso que he dado en mi vida. El fin de este proyecto es el comienzo de muchos otros más.

ÍNDICE DE CONTENIDO

| | |
|---|------------|
| LISTA DE FIGURAS | II |
| LISTA DE TABLAS | III |
| LISTA DE ANEXOS | IV |
| RESUMEN | V |
| ABSTRACT | VI |
| 1. INTRODUCCIÓN..... | 1 |
| 1.1. PREGUNTA DE INVESTIGACIÓN..... | 2 |
| 1.2. OBJETIVO GENERAL..... | 3 |
| 1.3. OBJETIVOS ESPECÍFICOS..... | 3 |
| 1.4. MARCO TEÓRICO..... | 3 |
| 2. METODOLOGÍA | 11 |
| 2.1. CONCEPTUALIZACIÓN DE LA IDEA..... | 11 |
| 2.2. PLANTEAMIENTO DEL PROBLEMA | 12 |
| 2.3. REVISIÓN DE LA LITERATURA | 13 |
| 2.4. VISUALIZACIÓN DEL ALCANCE DEL ESTUDIO..... | 15 |
| 2.5. ELABORACIÓN DE HIPÓTESIS Y DEFINICIÓN DE VARIABLES | 16 |
| 2.6. DISEÑO DE LA INVESTIGACIÓN | 17 |
| 2.7. DEFINICIÓN Y SELECCIÓN DE LA MUESTRA | 23 |
| 2.8. RECOLECCIÓN DE DATOS | 24 |
| 2.9. ANÁLISIS DE DATOS..... | 25 |
| 2.10. ELABORACIÓN DE RESULTADOS..... | 25 |
| 3. RESULTADOS Y DISCUSIÓN | 26 |
| 3.1. RESULTADOS | 26 |
| 3.2. DISCUSIÓN..... | 35 |
| 4. CONCLUSIONES | 36 |
| 5. REFERENCIAS BIBLIOGRÁFICAS..... | 38 |

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1.1 Clasificador médico | 4 |
| Figura 1.2 Etapas de un modelo auto supervisado [9] | 5 |
| Figura 1.3 Función de mapeo de identidad | 6 |
| Figura 1.4 Funciones de identidad aplicadas a imágenes de entrada..... | 7 |
| Figura 1.5 Aprendizaje tradicional vs Transfer Learning | 8 |
| Figura 1.6 Sol y lluvia..... | 9 |
| Figura 1.7 Modelo de aprendizaje supervisado [16]..... | 9 |
| Figura 1.8 Decaimiento del gradiente [13]..... | 10 |
| Figura 2.1 Fases de la investigación científica | 11 |
| Figura 2.2 Reconocimiento de rostros | 12 |
| Figura 2.3 Esquema de la construcción del dataset en [18] | 14 |
| Figura 2.4 WeatherNet [3]..... | 15 |
| Figura 2.5 Ejemplos de las clases del tipo de clima. | 16 |
| Figura 2.6 Arquitectura Resenet 50 | 17 |
| Figura 2.7 Fases del modelo de reconocimiento del clima | 18 |
| Figura 2.8 Ejemplo de clases de ImageNet | 19 |
| Figura 2.9 Predicción modelo pre entrenado con ImageNet | 19 |
| Figura 2.10 Ejemplo de imágenes Harvard Dataverse..... | 20 |
| Figura 2.11 Transformaciones en imágenes digitales..... | 21 |
| Figura 2.12 Preentrenamiento auto supervisado..... | 22 |
| Figura 2.13 Predicciones del modelo auto supervisado | 22 |
| Figura 2.14 Clases del entrenamiento supervisado..... | 23 |
| Figura 3.1 Diagrama de flujo modelo de reconocimiento del clima | 26 |
| Figura 3.2 Evolución del desempeño del modelo..... | 27 |
| Figura 3.3 Proceso preentrenamiento auto supervisado inicial | 29 |
| Figura 3.4 Proceso de preentrenamiento auto supervisado óptimo | 30 |
| Figura 3.5 Matriz de confusión..... | 32 |
| Figura 3.6 Proceso de fine tuning supervisado con preentrenamiento | 33 |
| Figura 3.7 Métrica de desempeño TOP3..... | 35 |

LISTA DE TABLAS

| | |
|--|----|
| Tabla 1 Resultados BiT | 28 |
| Tabla 2 Resultados obtenidos sin preentrenamiento..... | 32 |
| Tabla 3 Matriz de confusión multiclase | 33 |
| Tabla 4 Resultados obtenidos con preentrenamiento..... | 34 |

LISTA DE ANEXOS

| | |
|--|----|
| Anexo I – Probabilidades de predicción..... | 42 |
| Anexo II – Código Fuente..... | 50 |

RESUMEN

El reconocimiento automático del clima en imágenes digitales tiene importantes aplicaciones, tales como: control de tráfico terrestre y aéreo, vehículos autónomos, advertencias de seguridad vial, control de sembríos, entre otras. La construcción de modelos climáticos de aprendizaje profundo es un reto para la visión artificial a la hora de extraer las características determinísticas de cada tipo de clima. Por lo cual, varios trabajos relacionados han propuesto modelos basados en clasificadores binarios, que permiten catalogar entre clases positivas y negativas. Estos trabajos han reportado como principales dificultades que las clases no son completamente excluyentes y la necesidad de miles o millones de imágenes etiquetadas de cada clase para el entrenamiento supervisado de modelos de aprendizaje profundo.

Para abordar varias de las dificultades identificadas, en este trabajo de investigación proponemos aplicar los conceptos de “aprendizaje profundo auto supervisado”. El aprendizaje auto supervisado sigue las fases tradicionales del aprendizaje profundo y adiciona las etapas de preentrenamiento, transfer learning, y *fine tuning*. Las etapas de preentrenamiento son las que permiten reducir la cantidad de imágenes etiquetadas requeridas, mientras que las etapas posteriores permiten generalizar y mejorar el modelo.

La arquitectura de una red neuronal profunda de aprendizaje residual, RESNET por sus siglas en inglés, ha sido seleccionada para este trabajo basado en una investigación exhaustiva y varias pruebas prácticas que permitieron identificar los beneficios del uso de esta arquitectura en el campo del reconocimiento del clima. El método propuesto fue probado y ajustado hasta obtener métricas de desempeño del 93%.

Palabras clave: Reconocimiento automático del clima, aprendizaje auto supervisado, aprendizaje residual, transfer learning, fine tuning.

ABSTRACT

Automatic weather recognition from digital images has important applications, such as: land and air traffic control, autonomous vehicles, road safety warnings, crop control, among others. Building deep learning models for weather recognition is a challenge for computer vision when it comes to extracting the deterministic characteristics of each type of weather. Therefore, several related works have proposed models based on binary classifiers, which allow mapping between positive and negative classes. These works have reported as main difficulties that the classes are not completely exclusive and the need for thousands or millions of labeled images for each class required for training deep learning models.

To face several of the difficulties identified, in this research paper we propose to apply the concepts of "self-supervised deep learning". Self-supervised learning follows the traditional phases of deep learning and adds the stages of pre-training, transfer learning, and fine tuning. The pretraining stages are the ones that allow to reduce the number of labeled images required, while the later stages allow to generalize and improve the model.

The Residual Learning Network Architecture, RESNET, has been selected for this work based on exhaustive research and several practical tests that allow to identify the benefits of using this architecture in the field of weather recognition. The proposed method was tested and adjusted to achieve metrics of 93% in detection rates.

Keywords: Automatic weather recognition, self-supervised learning, residual learning, transfer learning, fine tuning.

1. INTRODUCCIÓN

El estado del clima es tan importante y visible que existen diversos medios para conocer los pronósticos del tiempo. Es una de las variables más importantes a la hora de decidir qué actividad realizar, además puede influir en nuestro estado de ánimo y las consecuencias que de él se derivan. Por otro lado, conocer los diferentes estados del clima ha permitido determinar la biodiversidad de las especies, ecosistemas y lugares naturales generados a partir de ellos. Lugares que pueden ser habitables o completamente inhabitables debido al clima [1].

Los avances tecnológicos han permitido inicialmente el diseño de sistemas analíticos que buscan predecir el estado del clima basado en información histórica sobre su comportamiento a lo largo del tiempo, utilizando variables como la temperatura, la presión atmosférica, vientos, humedad y precipitación.

Actualmente, con el avance del reconocimiento automático de imágenes, ha sido posible modelar el estado del clima en base a datos gráficos extraídos de imágenes etiquetadas de entrenamiento que contienen los diferentes estados del clima: nublado, neblinoso, lluvioso, soleado y amanecer [2]. Estos modelos han permitido resolver problemas que van desde propuestas tan simples como caminar o andar en bicicleta en una ciudad, a soluciones más complejas como los asistentes de conducción autónoma [3].

Según los autores en [4], las tormentas de lluvia, las tormentas de nieve y la niebla, son los tres tipos de clima extremos más estudiados con objetivos de seguridad vial. Estas condiciones conducen a una visibilidad y coeficiente de fricción de la carretera reducido, lo que resulta ser un peligro potencial. Por esa razón, reconocer automáticamente el clima es esencial para muchas aplicaciones, como advertencias de condiciones de tráfico en carreteras, automóviles de conducción autónoma, análisis climático para la agricultura, etc.

Varias soluciones propuestas para el reconocimiento automático del clima han encontrado limitaciones en sus resultados debido a la cantidad de imágenes que disponen para el entrenamiento supervisado de un modelo de aprendizaje profundo. Por esta razón, la mejora en los resultados obtenidos en trabajos previos está estrechamente relacionada al aumento de las imágenes de entrenamiento disponibles. Es decir, la

mayoría de los trabajos han enfocado sus esfuerzos en la creación y optimización de conjuntos de datos gráficos para el entrenamiento de modelos climáticos. Lamentablemente, varios de estos conjuntos de datos han sido generados para trabajos específicos por lo que no están disponibles públicamente para la construcción y entrenamiento de nuevos y mejorados modelos.

Entrenar un modelo de aprendizaje profundo con un enfoque totalmente supervisado requiere un esfuerzo temporal y económico excesivo para obtener datos correctos de entrenamiento. Por esta razón, es casi imposible aplicar aprendizaje profundo en escenarios donde los datos etiquetados son escasos, como en el planteamiento del problema del reconocimiento del clima. En la mayoría de los casos, entrenar un modelo de aprendizaje profundo requiere miles, cientos de miles, millones, incluso miles de millones de imágenes de entrenamiento para aprender representaciones precisas de las imágenes [5].

En el presente trabajo, a diferencia de los trabajos relacionados, hemos enfocado nuestros esfuerzos principalmente en el diseño de la arquitectura y fases de la red neuronal profunda que sea capaz de mejorar los resultados utilizando la mínima cantidad de imágenes climáticas etiquetadas de acceso público, mediante técnicas de aprendizaje auto supervisado.

Como resultado de este trabajo de investigación, proponemos un modelo de aprendizaje profundo con una red neuronal artificial RESNET-50, con dos etapas de preentrenamiento auto supervisado y una etapa de fine tuning, alcanzando con estas características el 93% en métricas de precisión en el reconocimiento del clima con solo 1148 imágenes etiquetadas de entrenamiento. Una mejora considerable tomando en cuenta las 16635 imágenes etiquetadas utilizadas para entrenar el modelo llamado "Extreme Weather" [4], que obtuvo los mejores resultados, 94%.

1.1. Pregunta de investigación

¿Pueden los métodos de aprendizaje profundo supervisado para el reconocimiento del clima ser optimizados mediante técnicas de preentrenamiento auto supervisado utilizando colecciones de datos aleatorias y no etiquetadas?

1.2. Objetivo general

Implementar un modelo de aprendizaje profundo auto supervisado para la clasificación y reconocimiento automático del clima a partir de imágenes digitales obtenidas de tres diferentes bases de datos públicas.

1.3. Objetivos específicos

- ✓ Revisar el estado del arte respecto de los modelos de aprendizaje profundo, aprendizaje auto supervisado, fine tuning y transferencia de conocimiento para el reconocimiento del clima en imágenes digitales.
- ✓ Definir las etapas, procesos, técnicas, arquitecturas y herramientas adecuadas para nuestro caso de estudio.
- ✓ Implementar un modelo auto supervisado mediante técnicas de preentrenamiento, transferencia de conocimiento y fine tuning para el reconocimiento del clima.
- ✓ Evaluar y optimizar el modelo para obtener resultados en métricas de exactitud top-1 y top-5 superiores a los obtenidos en el estado del arte con métodos completamente supervisados para el reconocimiento del clima.
- ✓ Documentar los resultados obtenidos y realizar un análisis comparativo con otras soluciones encontradas en el estado del arte.

1.4. Marco Teórico

Para abordar el problema del reconocimiento del clima haremos uso de técnicas de aprendizaje profundo auto supervisado. El aprendizaje profundo auto supervisado no es un enfoque nuevo, es un conjunto de técnicas tradicionales de aprendizaje de máquina optimizadas para el procesamiento de grandes cantidades de datos no etiquetados.

Este enfoque ha demostrado una gran capacidad en el aprendizaje automático en diversas tareas, principalmente en lo relacionado al procesamiento del lenguaje natural. Por lo que ahora buscamos aplicarlo al reconocimiento del clima en imágenes.

A continuación, se describe al aprendizaje profundo auto supervisado y cada una de las técnicas que lo conforman, así como también la definición de las métricas de desempeño utilizadas:

1.4.1. Aprendizaje Auto Supervisado

Entrenar un modelo de aprendizaje profundo con un enfoque completamente supervisado representa un gran reto, por lo que en años recientes se ha propuesto y probado un enfoque de aprendizaje profundo denominado “aprendizaje auto supervisado” [6]. Con este enfoque se ha conseguido obtener iguales e incluso mejores resultados que los enfoques de entrenamiento completamente supervisados en tareas de procesamiento de imágenes, debido a su versatilidad en el uso de datos no etiquetados en una fase preliminar denominada preentrenamiento.

El aprendizaje auto supervisado, fue inicialmente propuesto para tareas de procesamiento de lenguaje natural, debido a la gran cantidad de datos sueltos que se tiene para estas tareas. Sin embargo, este enfoque también ha sido probado en otros campos como: reconocimiento de actividad humana [7], exploración y desplazamiento [8], visión 360 [9], e incluso en tareas más críticas como clasificadores médicos [10], donde se propone un modelo de aprendizaje de 3 etapas:

1. Aprendizaje auto supervisado con imágenes naturales no etiquetadas (Imagenet usando SimCLR [11]).
2. Aprendizaje auto supervisado con imágenes médicas no etiquetadas usando Multi Instance Contrastive Learning (MICLe).
3. Fine Tuning supervisado con imágenes médicas etiquetadas

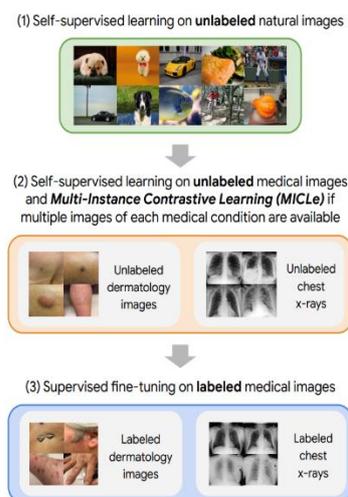


Figura 1.1 Clasificador médico

1.4.2. Etapas de un modelo de aprendizaje profundo auto supervisado

Un modelo de aprendizaje auto supervisado sigue las fases tradicionales de un modelo supervisado, pero adicionalmente agrega una o varias etapas de preentrenamiento que permiten ajustar los pesos de la red con imágenes que contienen etiquetas autogeneradas de acuerdo con una o varias transformaciones aplicadas. En el caso más sencillo, en una fase de preentrenamiento se puede entrenar un clasificador binario [9] que identifique si una señal ha sido previamente transformada o no. En la Figura 1.2 se puede observar un modelo clasificador de señales transformadas.

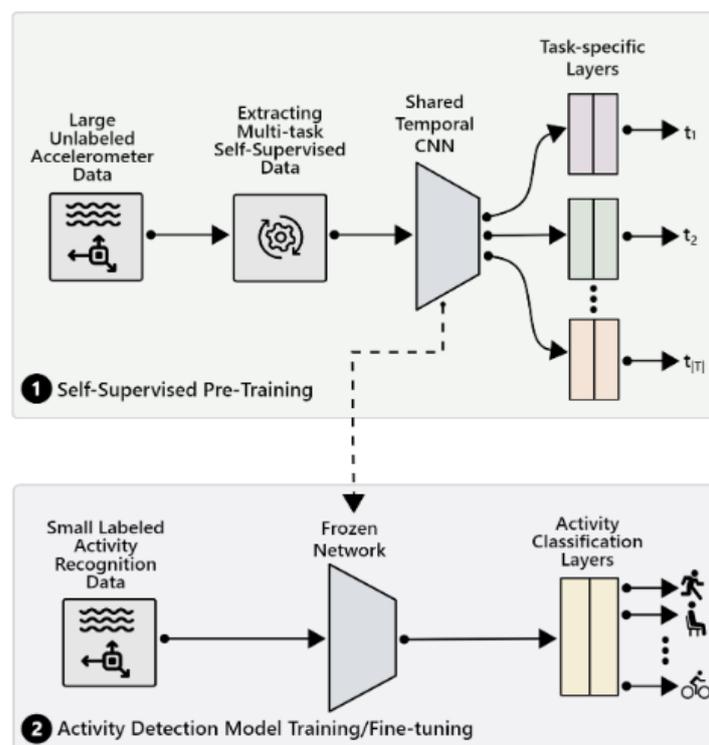


Figura 1.2 Etapas de un modelo auto supervisado [9]

En el preentrenamiento auto supervisado de este problema, clasificador de actividad humana, se aplica un conjunto de transformaciones T , el tipo y cantidad de transformaciones depende del problema y los resultados esperados. La red pre entrenada se mantiene fija para el posterior ajuste con señales etiquetas de las clases reales.

1.4.3. Residual Learning

Las redes residuales, Resnet, son un tipo de redes neuronales artificiales muy profundas que permiten reducir el problema del decaimiento del gradiente mientras se incrementa la profundidad de la red. Cuando las redes neuronales profundas aumentan en profundidad, mayor número de capas, también aumenta el problema del decaimiento del gradiente, por lo que la precisión se ve muy afectada a medida que las capas aumentan, en lugar de suceder lo contrario. Teóricamente, las capas más profundas extraerían mejores detalles de las imágenes con lo que el desempeño del modelo debería ser mejor, pero en realidad sucede lo contrario y esto no es debido a un sobreajuste del modelo sino a un problema inherente detectado en diversas pruebas realizadas en modelos de aprendizaje profundo. Por esta razón, las redes residuales introducen un término llamado “identity mapping” o mapeo de identidad [12] que agregan conexiones lineales entre las capas de la red para evitar el problema del gradiente. La siguiente imagen ilustra una función de identidad agregada a un bloque residual.

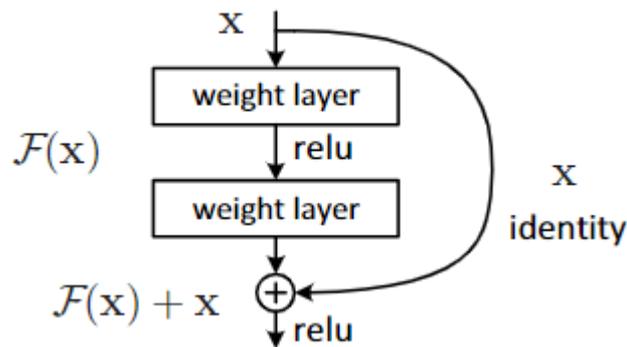


Figura 1.3 Función de mapeo de identidad

Estas conexiones lineales no incrementan complejidad computacional ni parámetros adicionales a la red. Por lo que son utilizadas únicamente para aumentar la profundidad de la red.

La red completa está compuesta por diversos bloques residuales, que son entrenados y ajustados utilizando funciones de gradiente descendente estocástico con propagación inversa.

En la Figura 1.4 se aplican funciones de identidad a una imagen de entrada x .

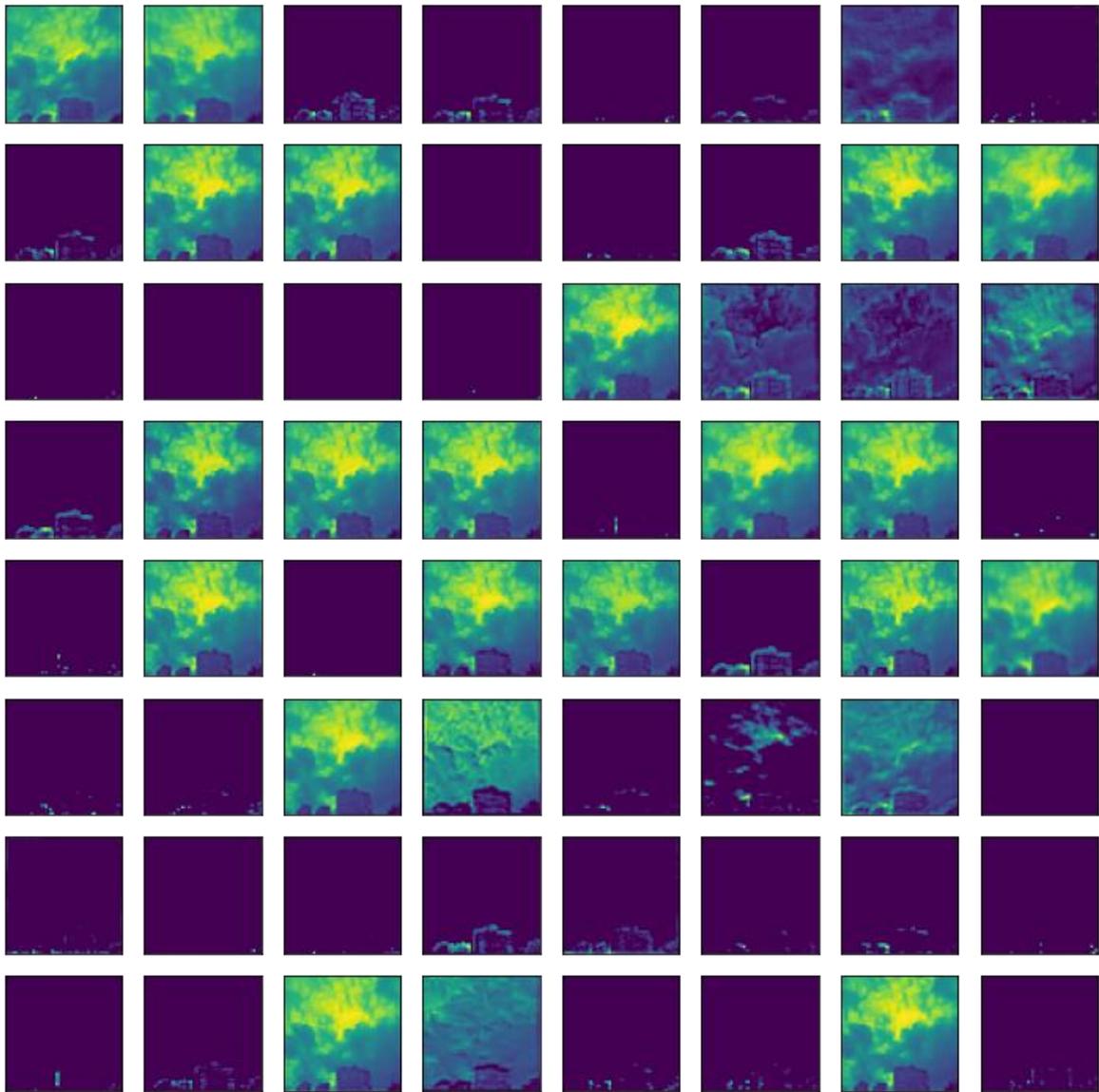


Figura 1.4 Funciones de identidad aplicadas a imágenes de entrada

Resenet ha alcanzado con éxito una profundidad máxima de 152 capas, 8 veces más profunda que las redes VGG “más profundas” previamente propuestas, pero con menor complejidad computacional, y con un error de 3.57% [13] evaluado sobre el dataset de pruebas del Imagenet [14].

1.4.4. Transfer Learning

El transfer Learning es un método mediante el cual se busca reutilizar los parámetros de una red previamente entrenada en otro problema de un dominio diferente con espacio de características y funciones de distribución completamente diferentes al del problema original para el cual la red fue entrenada [15]. En la Figura 1.5 se puede observar la diferencia entre un modelo aprendizaje tradicional y un modelo con transfer learning.

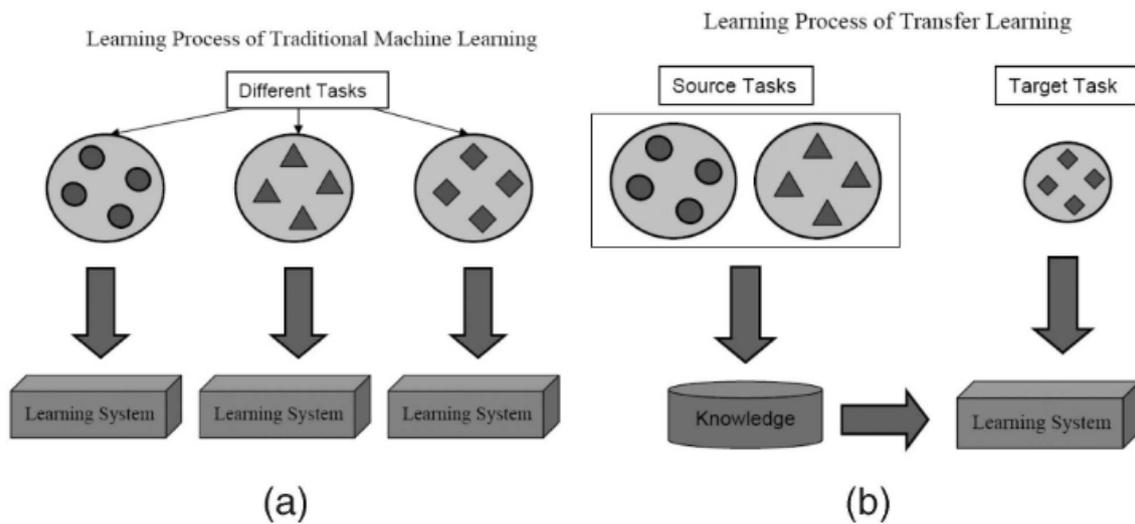


Figura 1.5 Aprendizaje tradicional vs Transfer Learning

1.4.5. Dificultades que se presentan en el reconocimiento automático del clima

En trabajos relacionados, se han reportado varias dificultades en las tareas de reconocimiento automático del clima, entre las más importantes tenemos:

- ✓ Las clases no siempre son mutuamente excluyentes, por lo que en varias situaciones podemos tener la presencia de múltiples clases en una misma imagen como en la Figura 1.6.



Figura 1.6 Sol y lluvia

- ✓ Entrenar un modelo de aprendizaje profundo completamente supervisado requiere de una gran cantidad de imágenes correctamente etiquetadas, por lo que la construcción de un conjunto de datos adecuado para el entrenamiento, validación y pruebas requiere de un gran esfuerzo, temporal y económico. La Figura 1.7 muestra el flujo de entrenamiento de una red neuronal.

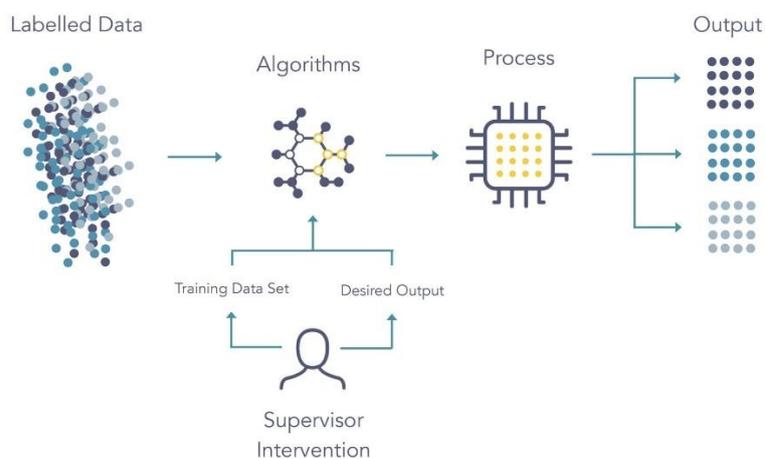


Figura 1.7 Modelo de aprendizaje supervisado [16]

- ✓ Entrenar un modelo de aprendizaje profundo está expuesto al problema del decaimiento del gradiente que se produce por el apilamiento excesivo de capas profundas lo que termina en un mayor error de entrenamiento y de pruebas [13].

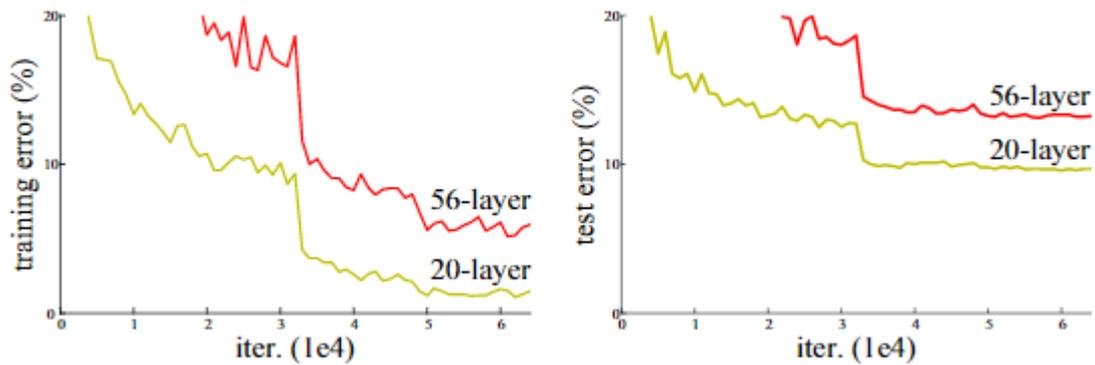


Figura 1.8 Decaimiento del gradiente [13].

1.5. Métricas de desempeño “Top N”

Las métricas de desempeño Top N tienen el mismo objetivo que la métrica de precisión tradicional pero enfocadas en las N mejores clases predichas por el modelo de inteligencia artificial.

Estas métricas tienen gran utilidad en los motores de sugerencias [17], donde dado un conjunto de entrada el motor puede sugerir N categorías similares mediante modelos de predicción. Sin embargo, en la actualidad han sido ampliamente utilizadas también en los modelos de predicción de imágenes para brindar mayor flexibilidad y adaptabilidad a los resultados.

La precisión TOP N hace referencia a la respuesta positiva con respecto a las N mejores respuestas del clasificador. Por ejemplo, si la clase real de una imagen es lluviosa y el modelo predice primero nublado, segundo neblinoso y tercero lluvioso, la respuesta será tomada como correcta para $N=3$ dado que la clase real se encuentra dentro de las 3 mejores predicciones del modelo.

2. METODOLOGÍA

Este trabajo de titulación se llevará a cabo mediante un proceso metódico y sistemático que permite brindar validez y confianza a la investigación. Así como también asegurar la reproducibilidad de los experimentos y los resultados.

El Método científico con un enfoque cuantitativo basado en la experimentación guiará el desarrollo del presente trabajo. En este capítulo se detallan las fases que permiten alcanzar los objetivos planteados para la investigación. En la Figura 2.1, se muestran las fases de la investigación científica.

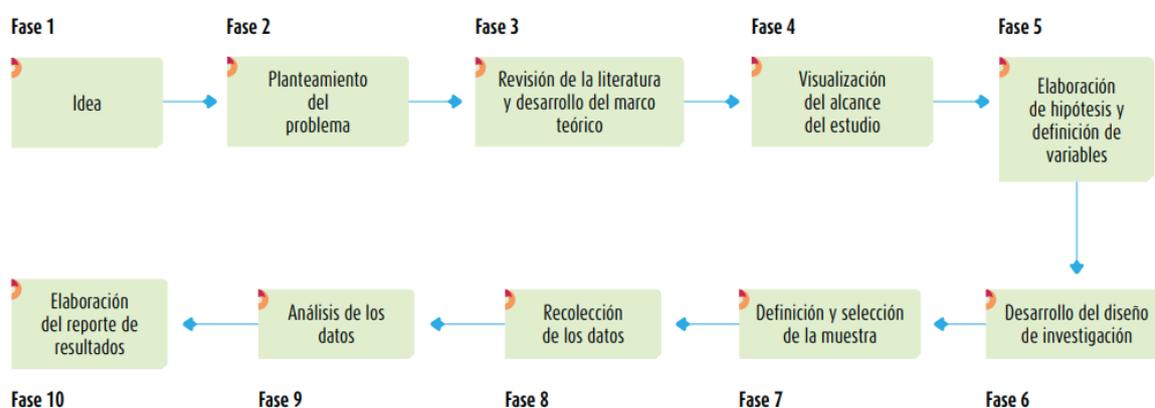


Figura 2.1 Fases de la investigación científica

2.1. Conceptualización de la idea

La idea de proponer un modelo de reconocimiento automático del clima no es nueva, sin embargo, de acuerdo con el estudio del estado del arte, se ha podido encontrar que varios trabajos basados en la misma idea han visto sus resultados limitados por la cantidad de datos de entrenamiento que han podido construir específicamente para sus trabajos. Esto impide que trabajos nuevos puedan hacer uso de los recursos obtenidos en investigaciones previas con el objetivo de mejorar los resultados y optimizar los modelos. Como se mencionó brevemente en la introducción y se verá con mayor detalle en la sección de resultados, los modelos entrenados en los trabajos relacionados han requerido de una gran cantidad de imágenes etiquetadas de entrenamiento, entre 10000 y 23865 imágenes como se ha reportado. Construir un dataset de estas dimensiones requiere de mucho esfuerzo económico y de tiempo. Por esta razón, en este trabajo de titulación mediante técnicas de preentrenamiento auto supervisado y fine tuning se busca

construir un modelo igual de bueno que los actuales, pero usando solo 1148 imágenes etiquetadas de entrenamiento que se encuentran disponibles en el internet.

2.2. Planteamiento del Problema

El problema principal está basado en los modelos actuales de reconocimiento del clima que utilizan enfoques completamente supervisados para el entrenamiento y aprendizaje. Estos enfoques a lo largo del tiempo han mejorado continuamente a tal punto de obtener resultados similares en tareas de procesamiento automático de imágenes en comparación con tareas realizadas manualmente. Sin embargo, tienen un inconveniente; y es que se requiere de una gran cantidad de imágenes etiquetadas para el entrenamiento, lo que complica la implementación en entornos donde estos datos son escasos tal como el caso del reconocimiento del clima. Por esta razón, en este campo las métricas de desempeño obtenidas no han alcanzado resultados tan buenos como en otras tareas como el reconocimiento e identificación de rostros, Figura 2.2.



Figura 2.2 Reconocimiento de rostros

La generación de datos sintéticos mediante data augmentation o GANs probablemente ayudaría a mejorar los resultados, pero requeriría de una mayor cantidad de recursos computacionales, lo que nuevamente se traduce en otro problema. Por esta razón en este trabajo se busca balancear los resultados con los recursos e imágenes requeridas y se deja para una próxima investigación las ventajas y mejoras que se pueden obtener mediante el uso de datos sintéticos.

2.3. Revisión de la literatura

En la revisión de la literatura se consideran trabajos relacionados al reconocimiento automático del clima que han sido escritos en inglés y año de publicación desde 2010 al 2022.

En el estado del arte se han encontrado varios enfoques y metodologías en el campo del reconocimiento del clima, tal como; aprendizaje multitarea [18], donde se propone complementar la predicción con información climática auxiliar para poder determinar una definición comprensiva del clima, mediante tareas de clasificación categórica (sol, lluvia, etc.) y segmentación de señales meteorológicas como nivel de azul del cielo y nivel de blancura de las nubes. Este trabajo limita su alcance a dos clases, clasificación binaria, y su conjunto de datos está compuesto por 10k imágenes etiquetadas de entrenamiento, 5k por cada clase.

En [2], se propone un modelo de aprendizaje de diccionario categórico-específico y aprendizaje de kernel múltiple. El modelo extrae múltiples características de climas específicos y genera diccionarios basados en estas características. Para extraer el conjunto de características más adecuado los autores utilizan un algoritmo de aprendizaje de kernel múltiple para determinar una combinación lineal óptima de kernels de características. Para este trabajo los autores generaron su propio conjunto de datos compuesto de 20k imágenes meteorológicas.

Los autores en [19], complementan los datos gráficos del clima con propiedades meteorológicas y espaciales; hora, ubicación, anotaciones textuales y elevación. Toda esta información es obtenida de diferentes fuentes de datos disponibles en internet. El conjunto de datos fue construido en etapas múltiples para descartar imágenes que no cuentan con la información completa, de esta manera se obtiene un total 183798 completamente etiquetadas.

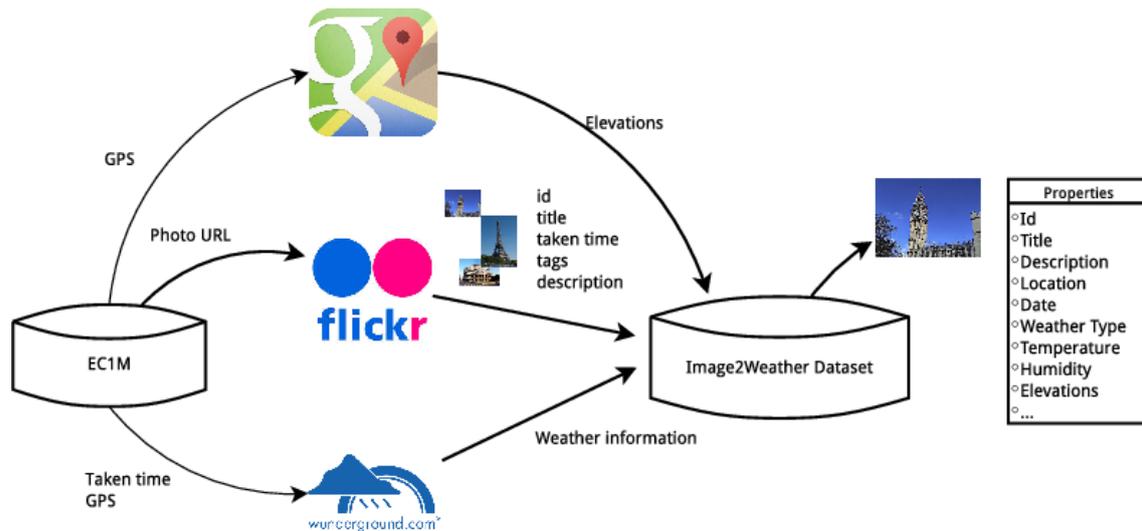


Figura 2.3 Esquema de la construcción del dataset en [19]

El problema del reconocimiento de climas extremos con objetivos de seguridad vial puede ser resuelto mediante el diseño e implementación de redes neuronales convolucionales como en [4], aplicando operaciones de convolución y reducción de dimensionalidad (downsampling) para obtener características más complejas de las imágenes. El conjunto de datos de este trabajo ha sido generado por los autores con un total de 16635 imágenes obtenidas del internet de acuerdo con los requerimientos del trabajo.

En [3], se propone el framework WeatherNet, que está compuesto por 4 redes profundas paralelas que reconocen el clima y las condiciones visuales de imágenes meteorológicas. Las redes son:

1. NightNet: detecta las diferencias entre el día y la noche.
2. GlareNet: detecta la luminosidad de las imágenes.
3. PrecipitationNet: detecta condiciones de lluvia, nieve o claridad.
4. FogNet: detecta la presencia de niebla en las imágenes.

Las redes 2 y 4 son entrenadas como clasificadores binarios, mientras que las redes 1 y 3 son entrenados como clasificadores multiclase. El conjunto de datos está compuesto por 23865 imágenes para entrenamiento y validación.

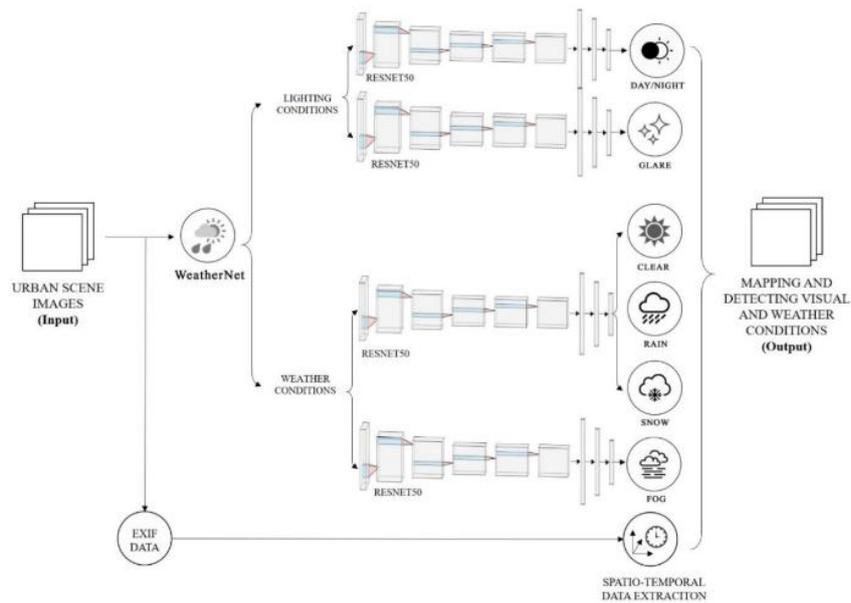


Figura 2.4 WeatherNet [3]

2.4. Visualización del alcance del estudio

El reconocimiento automático del clima es un campo de estudio muy amplio, por ello existe la necesidad de limitar el alcance en base a los objetivos planteados. Por lo que enfocamos este trabajo en el reconocimiento de 3 clases de clima; nublado, lluvioso y soleado. Además, se agregará una cuarta clase, llamada “other”, en la cual se agrupan los demás estados climáticos. Los resultados obtenidos y el rendimiento del modelo serán reportados en medidas como la exactitud top-1 y top-5, con las que han sido evaluados la mayoría de los algoritmos en el campo del procesamiento automático de imágenes, en nuestro caso calcularemos específicamente la métrica top-3 por el número de clases de nuestro problema. En la Figura 2.5, se puede observar unos ejemplos de las clases que van a considerarse.



Figura 2.5 Ejemplos de las clases del tipo de clima.

2.5. Elaboración de hipótesis y definición de variables

La hipótesis que se plantea es: Los métodos tradicionales de aprendizaje profundo supervisado para el reconocimiento del clima pueden ser optimizados mediante técnicas de preentrenamiento auto supervisado utilizando colecciones aleatorias y no etiquetadas de datos.

En este trabajo, abarcaremos el problema de reconocimiento del clima enfocados en tres tipos de clima: soleado, lluvioso y nublado. Para extraer información relevante de las imágenes, pre entrenamos una red neuronal "muy profunda", siguiendo una implementación ResNet-50 [13], Figura 2.6, agregando módulos de regularización [20], para la extracción de características visuales complementarias para el preentrenamiento auto supervisado y fine tuning supervisado. Se considerarán las etapas en [10] para el ajuste del modelo.

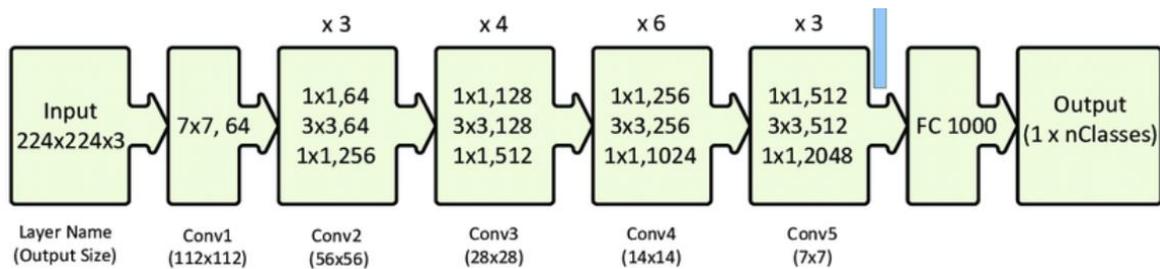


Figura 2.6 Arquitectura Resenet 50

Los parámetros de la red se mantienen fijos a lo largo de todos los preentrenamientos; batch size de 16, Learning rate inicial de 0,00009 que se ajusta en función del avance del entrenamiento a razón del 10% de reducción en cada ajuste y momentum de 0.9. Estos valores fueron previamente calculados en base a pruebas y capacidad computacional disponible. Para la optimización de la red se utiliza el algoritmo de gradiente descendente estocástico y entropía cruzada como función de pérdida. Los datos son divididos 80% y 20% para entrenamiento y validación respectivamente.

2.6. Diseño de la investigación

El entrenamiento del modelo auto supervisado para el reconocimiento del clima propuesto en este trabajo de investigación está dividido en 2 fases de preentrenamiento; preentrenamiento supervisado con ImageNet [14], preentrenamiento auto supervisado con Harvard Dataverse [21], y una fase de fine tuning con imágenes etiquetadas de WeatherNet [3].

A continuación, en la Figura 2.7, se muestra un esquema general de las fases para el entrenamiento del modelo de reconocimiento automático del clima propuesto en este trabajo de investigación.

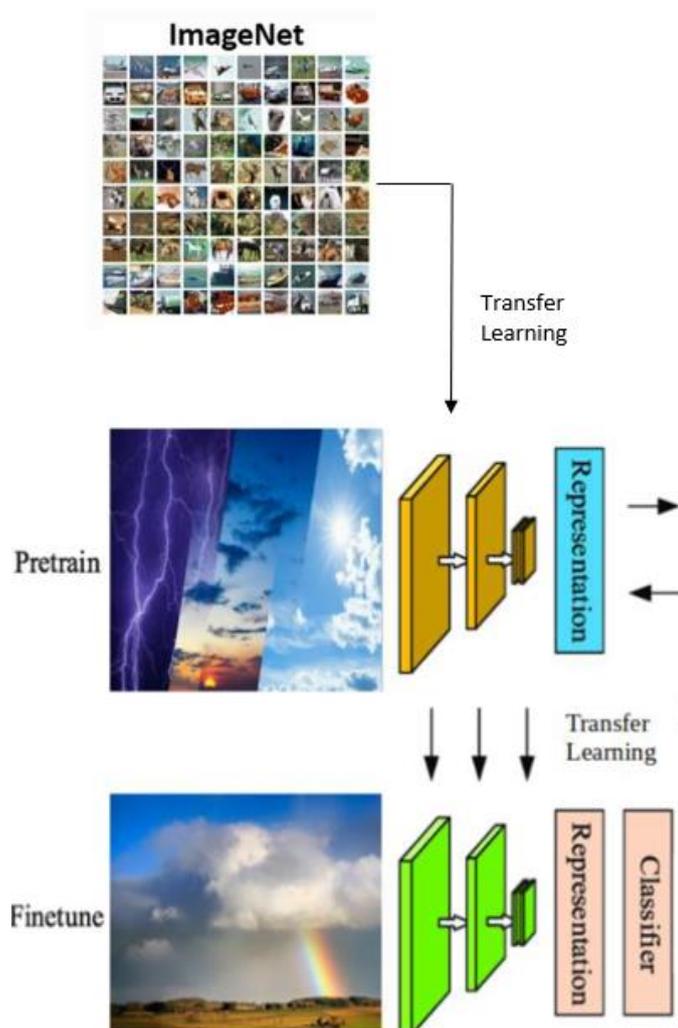


Figura 2.7 Fases del modelo de reconocimiento del clima

2.6.1. Preentrenamiento supervisado con ImageNet

Big Transfer (BiT) [22], es un método de preentrenamiento de modelos de clasificación de imágenes basado en grandes conjuntos de datos como ImageNet-21k compuesto por 14 millones de imágenes divididas en 21834 clases. La Figura 2.8 muestra algunas clases de ejemplo incluidas en el conjunto de datos de ImageNet.

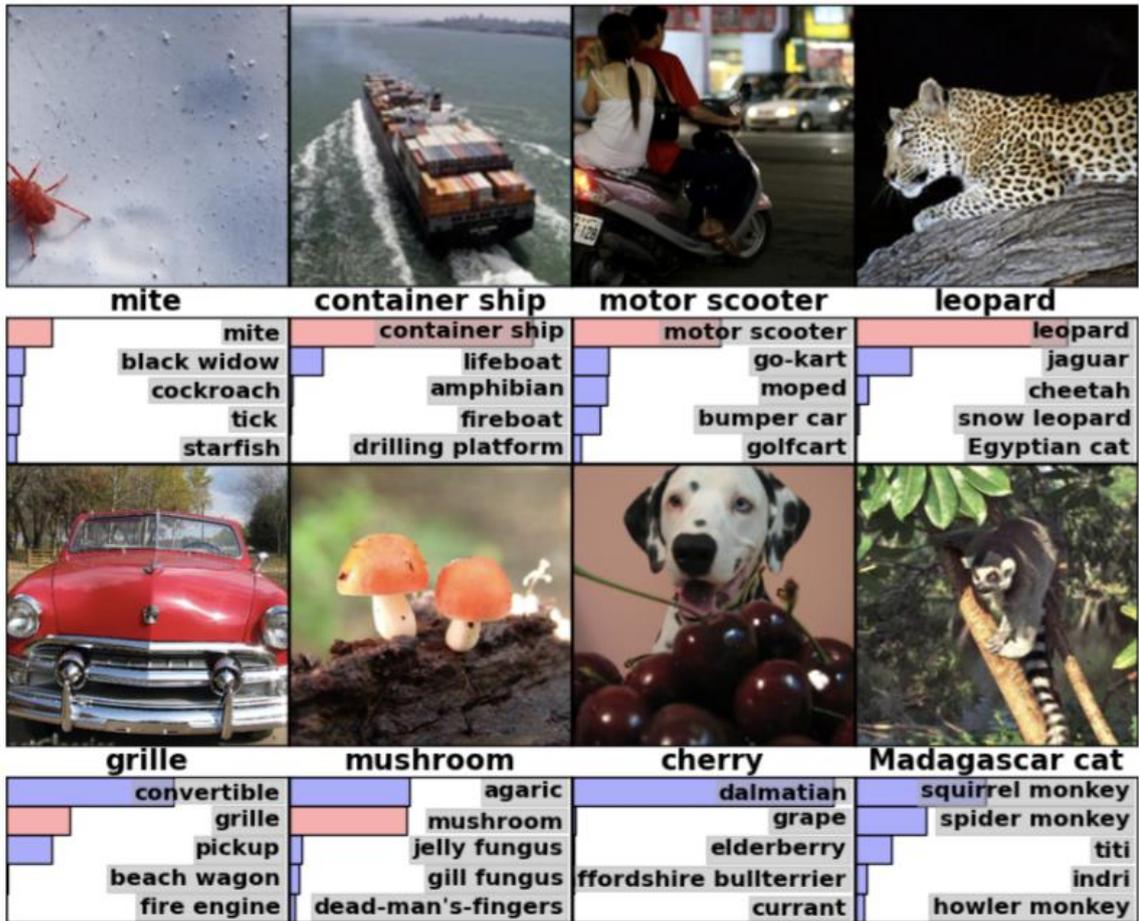


Figura 2.8 Ejemplo de clases de ImageNet

Con BiT se ha alcanzado una precisión Top-1 de 87.5% en el reto ILSVRC-2012 [23] y 99.4% en el conjunto de datos del CIFAR-10, compuesto por 60000 imágenes divididas en 10 clases. En la Figura 2.9 se puede observar una predicción del modelo entrenado con ImageNet basado en BiT.

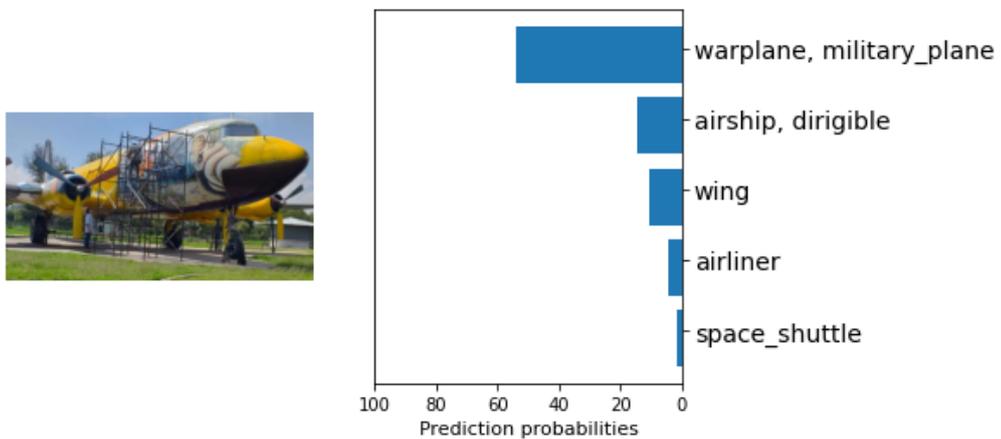


Figura 2.9 Predicción modelo pre entrenado con ImageNet

Mediante las técnicas de transferencia de conocimiento previamente expuestas es posible continuar a la siguiente fase de preentrenamiento.

2.6.2. Preentrenamiento auto supervisado con Harvard Dataverse

En esta fase hacemos uso del conjunto de datos de imágenes construida para el proyecto de monitoreo de vías mediante cámaras de circuito cerrado instaladas a lo largo de carreteras de Polonia [24]. Este conjunto de datos está compuesto por 3300000 imágenes no etiquetadas recolectadas en las carreteras. En la Figura 2.10 se puede observar unos ejemplos de imágenes del Harvard Dataverse.



Figura 2.10 Ejemplo de imágenes Harvard Dataverse

A dichas imágenes aplicamos 3 diferentes transformaciones más la imagen sin transformación con un total de 4 clases; Gaussian Noise, Glass Blur, Not transformed, Zoom Blur, mismas que serán consideradas para la generación de las auto etiquetas requeridas para el entrenamiento del modelo auto supervisado de esta fase. En la Figura 2.11 se pueden observar varias transformaciones que se pueden aplicar a una imagen de entrada.

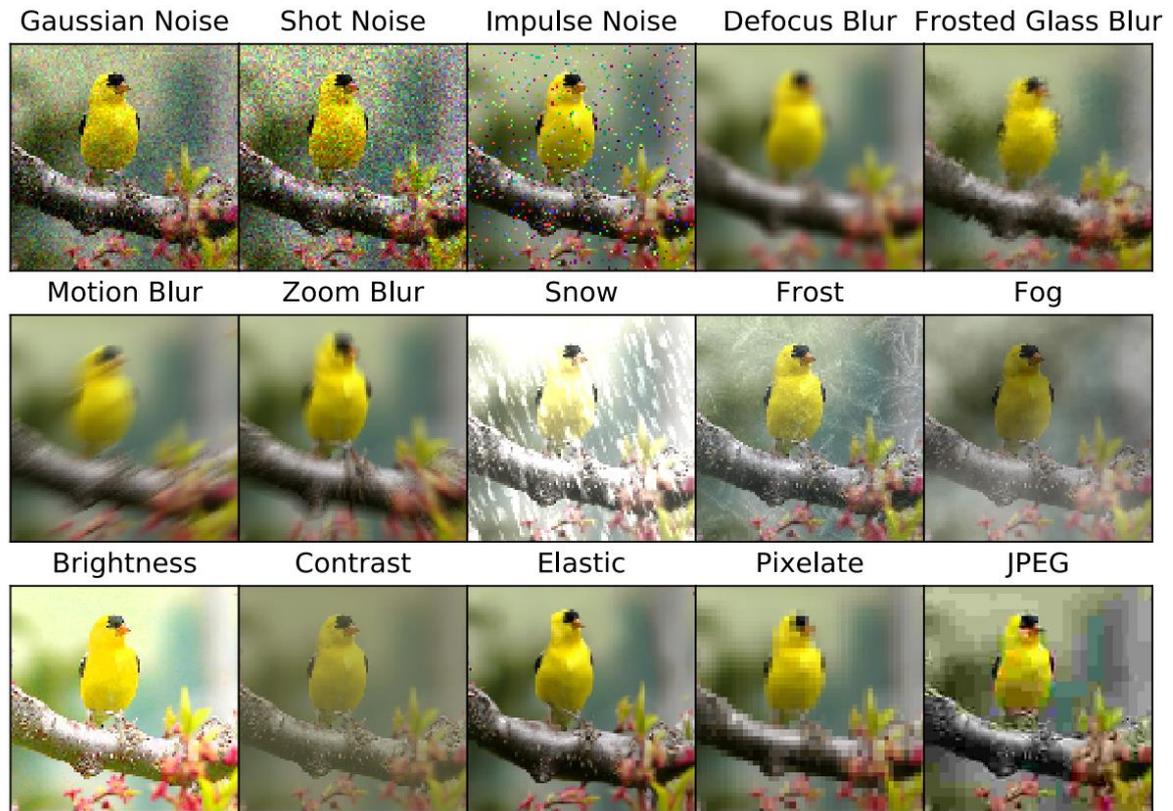


Figura 2.11 Transformaciones en imágenes digitales

En el preentrenamiento auto supervisado se usa la misma arquitectura de la red, RESNET-50, con 150 épocas de entrenamiento, batch size de 16, Learning rate inicial de 0,00009 que se ajusta en función del avance del entrenamiento a razón del 10% de reducción en cada ajuste.

La precisión TOP-1 obtenida al finalizar el preentrenamiento auto supervisado es del 98%. En las Figura 2.12 y Figura 2.13, se puede observar el avance del preentrenamiento y los resultados de precisión alcanzados.

```

2022-10-30 09:05:08.643680: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror:
libcudart.so.11.0: cannot open shared object file: No such file or directory
2022-10-30 09:05:08.643704: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2022-10-30 09:05:12.289032: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.
1: cannot open shared object file: No such file or directory; LD_LIBRARY_PATH: /home/fullstack-developer/.local/lib/python3.9/site-packages/cv2/./../lib64:
2022-10-30 09:05:12.289055: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] Failed call to cuInit: UNKNOWN ERROR (303)
2022-10-30 09:05:12.289068: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host
(Fullstackdeveloper-desktop): /proc/driver/nvidia/version does not exist
2022-10-30 09:05:12.289215: I tensorflow/core/platform/cpu_feature_guard.cc:151] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
WARNING:tensorflow:No training configuration found in save file, so the model was 'not' compiled. Compile it manually.
Found 22597 files belonging to 4 classes.
Using 18078 files for training.
Found 22597 files belonging to 4 classes.
Using 4519 files for validation.
1129/1129 [=====] - 1575s 14s/step - loss: 0.0623 - accuracy: 0.9846 - val_loss: 0.0183 - val_accuracy: 0.9965
2022-10-30 13:28:05.303259: W tensorflow/python/util/util.cc:368] Sets are not currently considered sequences, but this may change in the future, so consider
avoiding using them.

2022-10-30 09:05:08.643680: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror:
libcudart.so.11.0: cannot open shared object file: No such file or directory
2022-10-30 09:05:08.643704: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2022-10-30 09:05:12.289032: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.
1: cannot open shared object file: No such file or directory; LD_LIBRARY_PATH: /home/fullstack-developer/.local/lib/python3.9/site-packages/cv2/./../lib64:
2022-10-30 09:05:12.289055: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] Failed call to cuInit: UNKNOWN ERROR (303)
2022-10-30 09:05:12.289068: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host
(Fullstackdeveloper-desktop): /proc/driver/nvidia/version does not exist
2022-10-30 09:05:12.289215: I tensorflow/core/platform/cpu_feature_guard.cc:151] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2022-10-30 13:28:05.303259: W tensorflow/python/util/util.cc:368] Sets are not currently considered sequences, but this may change in the future, so consider
avoiding using them.
WARNING:absl:Found untraced functions such as head/dense_layer_call_fn, head/dense_layer_call_and_return_conditional_losses, re_lu_48_layer_call_fn,
re_lu_48_layer_call_and_return_conditional_losses while saving (showing 4 of 4). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: /media/fullstack-developer/0fab26ec-e67c-4a43-93fc-0ecfd4363f86/developer/workspace/big transfer/model_multi_weather_ssl/
assets
INFO:tensorflow:Assets written to: /media/fullstack-developer/0fab26ec-e67c-4a43-93fc-0ecfd4363f86/developer/workspace/big transfer/model_multi_weather_ssl/
assets

In [2]:

```

Figura 2.12 Preentrenamiento auto supervisado

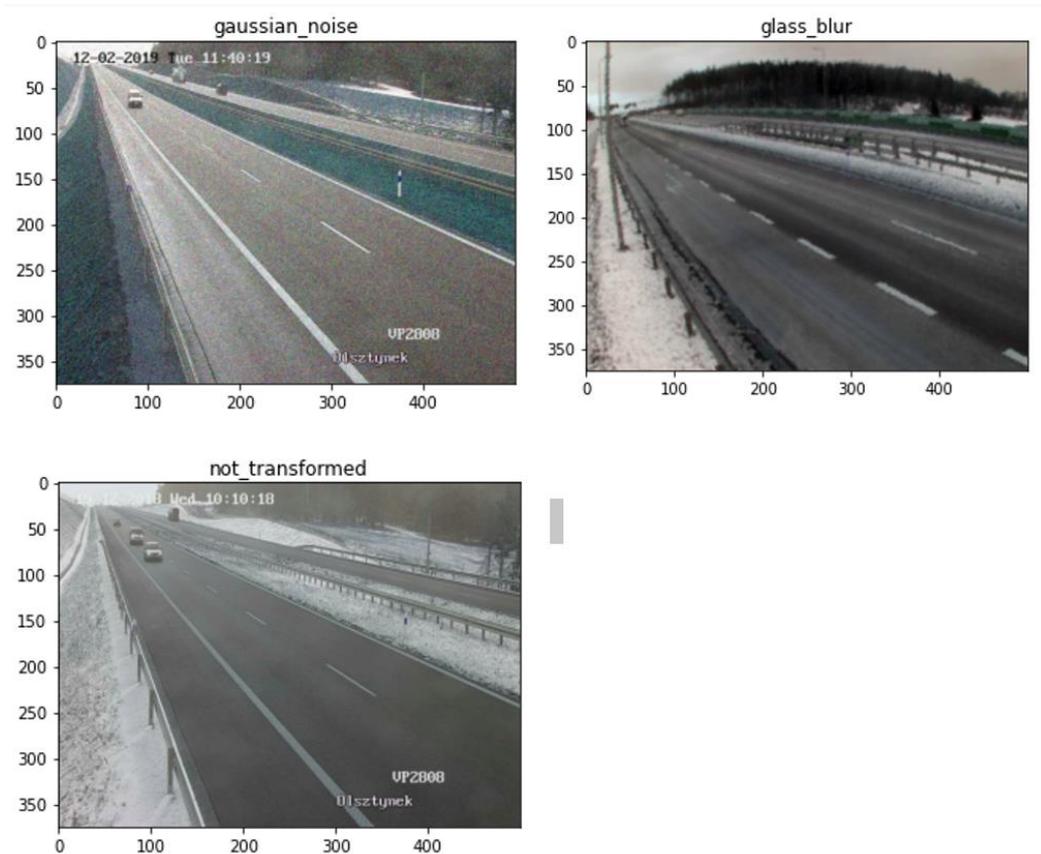


Figura 2.13 Predicciones del modelo auto supervisado

2.6.3. Fine tuning supervisado con Weather Dataset

En esta etapa final, al igual que en las etapas anteriores mediante transfer learning reentrenamos la red para que se ajuste a las clases reales del problema del reconocimiento del clima. En la Figura 2.14, se muestra algunos ejemplos de las clases usadas en esta fase de fine tuning supervisado.



Figura 2.14 Clases del entrenamiento supervisado

Para esta fase los parámetros de la red se mantuvieron constantes con la única variación de 15 épocas de entrenamiento.

2.7. Definición y selección de la muestra

Aquí, segmentamos los tipos de imágenes que necesitamos para llevar a cabo nuestros experimentos. Considerando; calidad, cantidad y contenido necesario.

Para la primera etapa de preentrenamiento con ImageNet, la red neuronal ha sido entrenada con el total de imágenes disponibles, es decir, 14 millones de imágenes divididas en 21834 clases.

Para el preentrenamiento auto supervisado se han tomado un total de 22596 imágenes aleatorias del total de 3300000 que contiene el Harvard Dataverse. Posteriormente, estas imágenes serán aplicadas una determinada transformación; Gaussian Noise, Glass Blur, Not transformed, Zoom Blur. Las transformaciones tienen igual probabilidad de ocurrencia, con lo que busca obtener un conjunto de datos balanceado entre todas las clases objetivo.

Finalmente, para el entrenamiento supervisado, segmentamos un total de 1148 imágenes etiquetadas con sus respectivas clases pertenecientes al conjunto de datos Weather Dataset.

Para la validación, se ha configurado un conjunto de datos de imágenes aleatorias obtenidas del internet con sus respectivas etiquetas, pertenecientes a las clases objetivo, mediante un algoritmo desarrollado con este objetivo específico. El conjunto de datos obtenido está compuesto por 280 imágenes en total distribuidas de manera equitativa entre las 4 clases objetivo.

Todos los conjuntos de datos originales fueron preprocesados con el objetivo de formatearlos de acuerdo con las características de los conjuntos de datos de tensorflow. Ver Anexos.

2.8. Recolección de datos

Los conjuntos de datos utilizados en todas las fases han sido extraídos de bases de datos públicas disponibles en las páginas web oficiales de los respectivos proyectos.

- ✓ ImageNet: Extraído de la página oficial del proyecto. [14]

- ✓ Harvard Dataverse: Los datos fueron extraídos de la página web oficial del proyecto perteneciente a la Universidad de Harvard [25].
- ✓ Weather Dataset: El conjunto de datos de imágenes del clima desarrollado para el proyecto WeatherNet [3], se encuentra publicado en un repositorio público de datos llamado Kaggle [26]. En dicho repositorio se puede encontrar conjuntos de datos de entrenamiento para problemas enfocados en diferentes contextos.
- ✓ Conjunto de datos de validación: Para construir el conjunto de datos de validación se desarrolló un algoritmo de web scraping para búsqueda y extracción de imágenes del internet [27]–[29] junto a sus respectivas etiquetas.

2.9. Análisis de datos

Ejecutamos una serie de experimentos que permiten obtener los resultados de acuerdo con las características extraídas de los conjuntos de datos y los ajustes de los parámetros de la red. Estos experimentos fueron optimizados y ejecutados hasta obtener los resultados deseados.

2.10. Elaboración de resultados

En los entrenamientos y pruebas se obtuvieron varios resultados referentes a las fases en que se realizaron las pruebas. Estos resultados son descritos con mayor detalle en el capítulo 3 de la investigación.

Los resultados también muestran las grandes diferencias que existieron entre los modelos climáticos pre entrenados y los sin preentrenamiento.

3. RESULTADOS Y DISCUSIÓN

El proceso cíclico de pruebas, análisis de resultados y reajuste de los parámetros de la red, se repitió hasta obtener un rendimiento que cumpla con los objetivos. En la Figura 3.1, se muestra el diagrama de flujo del entrenamiento del modelo para el reconocimiento automático del clima.

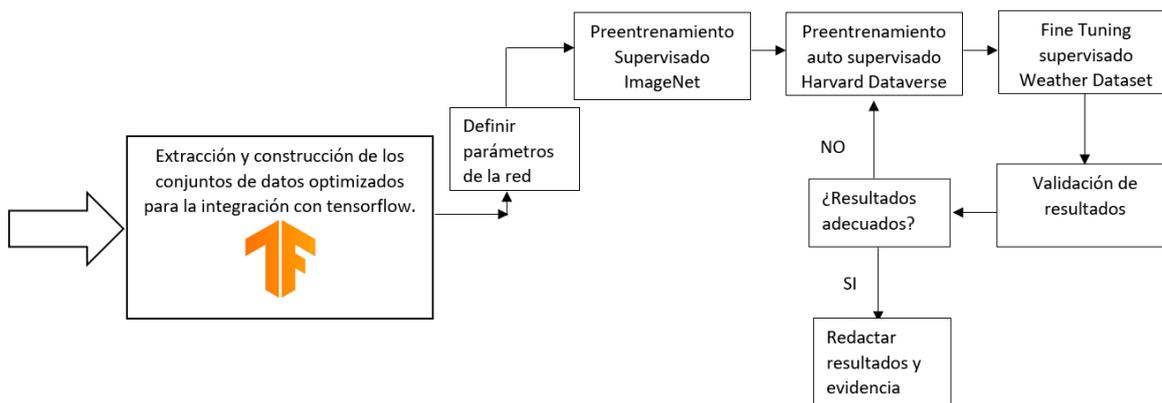


Figura 3.1 Diagrama de flujo modelo de reconocimiento del clima

3.1. Resultados

A continuación, se muestran los resultados obtenidos en cada una de las fases de preentrenamiento y fine tuning.

Preentrenamiento Supervisado ImageNet

La red pre entrenada con ImageNet para transferencia de conocimiento es un trabajo desarrollado en los laboratorios de Google por el Brain Team [22].

El modelo ha sido entrenado con el Dataset ImageNet-21k que contiene 14 millones de imágenes divididas en 21834 clases.

Los resultados obtenidos refieren una precisión TOP1 del 87.5% en ImageNet y 99.4% en CIFAR-10.

En la Figura 3.2, se puede observar la precisión de los resultados de transferencia luego de entrenar en subconjuntos de datos aleatorios balanceados.

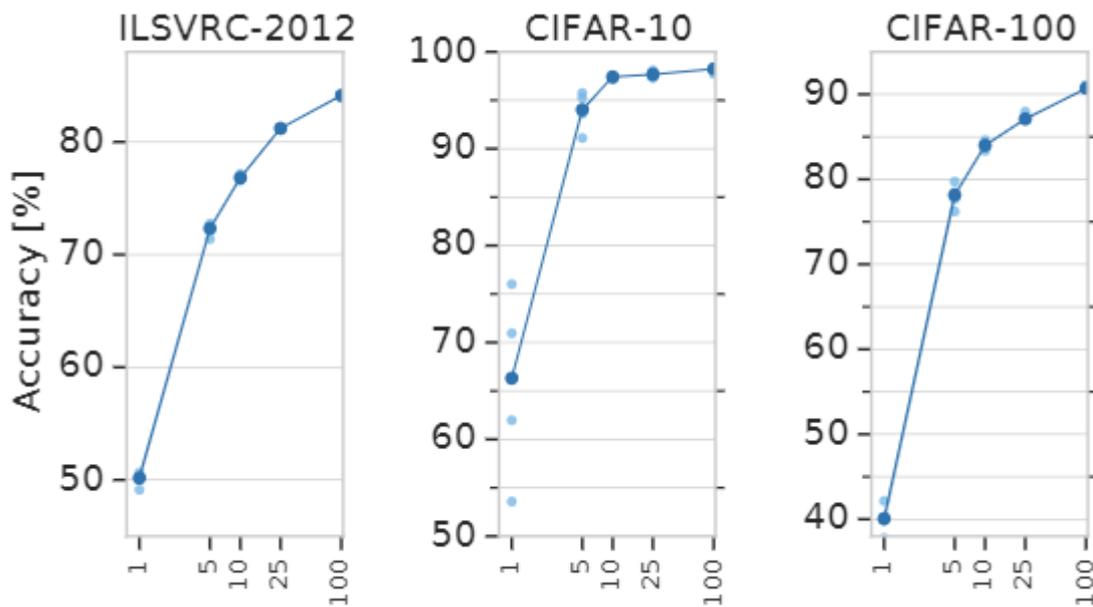


Figura 3.2 Evolución del desempeño del modelo

La evaluación y pruebas se las realizan sobre diferentes conjuntos de datos usando el mismo modelo pre entrenado y el mismo conjunto de hiper parámetros en todas las pruebas. Se han realizado 3 ejecuciones de fine tuning para optimizar la media y desviación estándar. Estos resultados demuestran una gran mejora respecto a las pruebas similares realizadas con modelos que no han sido previamente pre entrenados.

En la **¡Error! No se encuentra el origen de la referencia.**, se pueden observar los resultados obtenidos por el modelo BiT-L en diferentes subconjuntos de datos, las métricas muestran la mediana y su desviación estándar, comparado con un modelo generalista del estado del arte (SOTA).

Tabla 1 Resultados BiT

| | BiT-L | Generalist SOTA |
|-----------------|---------------------|-----------------|
| ILSVRC-2012 | 87.54 ± 0.02 | 86.4 |
| CIFAR-10 | 99.37 ± 0.06 | 99.0 |
| CIFAR-100 | 93.51 ± 0.08 | 91.7 |
| Pets | 96.62 ± 0.23 | 95.9 |
| Flowers | 99.63 ± 0.03 | 98.8 |
| VTAB (19 tasks) | 76.29 ± 1.70 | 70.5 |

En el subconjunto de datos ImageNet (ILSVRC-2012) el modelo BiT-L obtiene una precisión de 87.54% y una desviación estándar de 0.02. Esto representa una significativa mejora respecto al modelo del estado del arte que alcanza una precisión del 86.4%.

El modelo BiT-L, también fue probado en otros subconjuntos de datos como el CIFAR-10 y CIFAR-100 [30], Pets, Flowers y VTAB.

El conjunto de datos VTAB [31], consiste en 19 tareas visuales diversas, cada una con 1000 ejemplos de entrenamiento. En este subconjunto de datos el modelo BIT-L alcanza una precisión del 76.29%. A pesar de esta no ser una métrica óptima, representa una gran mejora frente a su antecesora que alcanzó el 70.5%.

Preentrenamiento auto supervisado Harvard Dataverse

El objetivo del preentrenamiento con el conjunto de datos Harvard Dataverse es ajustar el modelo con imágenes relacionadas al clima. Sin embargo, estas imágenes aún no tienen las etiquetas objetivo. Por lo que, como se explicó en el capítulo anterior, el algoritmo de preentrenamiento auto supervisado generará sus propias etiquetas de acuerdo con las transformaciones que se apliquen a las imágenes. Con esto buscamos usar la mayor cantidad de imágenes posibles sin la limitación del etiquetado manual que es una de las mayores restricciones en el entrenamiento de modelos supervisados de inteligencia artificial.

La fase de preentrenamiento auto supervisado fue realizada por varias ocasiones hasta obtener resultados óptimos. Las pruebas se realizaron variando los parámetros de la red,

cantidad de imágenes y transformaciones aplicadas. En la Figura 3.3, se puede ver el proceso del preentrenamiento auto supervisado en el primer intento.

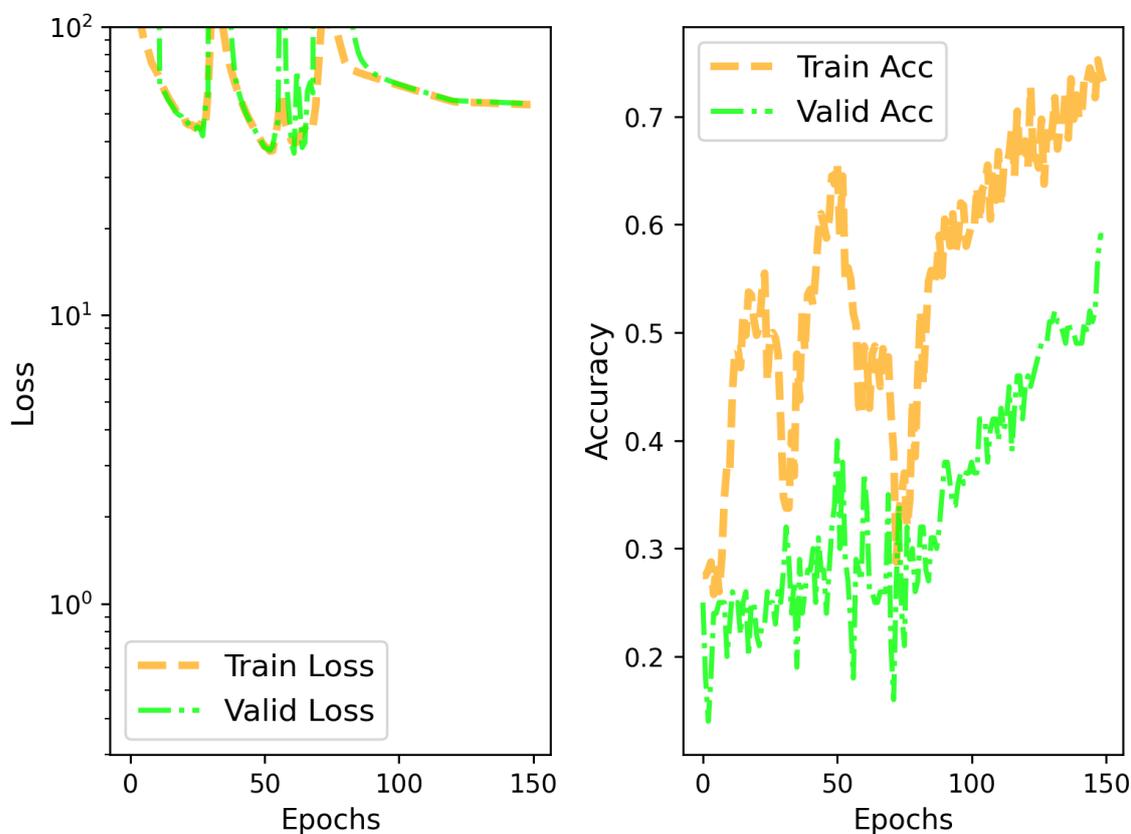


Figura 3.3 Proceso preentrenamiento auto supervisado inicial

Luego de varios entrenamientos con prueba y error llegamos a un modelo óptimo para este trabajo de investigación. En la Figura 3.4, se puede observar el proceso del preentrenamiento auto supervisado en el que obtuvimos los mejores resultados.

```

2022-10-30 09:05:08.643680: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dLError:
libcudart.so.11.0: cannot open shared object file: No such file or directory
2022-10-30 09:05:08.643704: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dLError if you do not have a GPU set up on your machine.
2022-10-30 09:05:12.289032: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcuda.so.1'; dLError: libcuda.so.
1: cannot open shared object file: No such file or directory; LD_LIBRARY_PATH: /home/fullstack-developer/.local/lib/python3.9/site-packages/cv2/./../lib64:
2022-10-30 09:05:12.289055: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit: UNKNOWN ERROR (303)
2022-10-30 09:05:12.289068: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host
(Fullstackdeveloper-desktop): /proc/driver/nvidia/version does not exist
2022-10-30 09:05:12.289215: I tensorflow/core/platform/cpu_feature_guard.cc:151] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
WARNING:tensorflow:No training configuration found in save file, so the model was *not* compiled. Compile it manually.
Found 22597 files belonging to 4 classes.
Using 18078 files for training.
Found 22597 files belonging to 4 classes.
Using 4519 files for validation.
1129/1129 [=====] - 1575s 14s/step - loss: 0.0623 - accuracy: 0.9846 - val_loss: 0.0183 - val_accuracy: 0.9965
2022-10-30 13:28:05.303259: W tensorflow/python/util/util.cc:368] Sets are not currently considered sequences, but this may change in the future, so consider
avoiding using them.

2022-10-30 09:05:08.643680: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dLError:
libcudart.so.11.0: cannot open shared object file: No such file or directory
2022-10-30 09:05:08.643704: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dLError if you do not have a GPU set up on your machine.
2022-10-30 09:05:12.289032: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcuda.so.1'; dLError: libcuda.so.
1: cannot open shared object file: No such file or directory; LD_LIBRARY_PATH: /home/fullstack-developer/.local/lib/python3.9/site-packages/cv2/./../lib64:
2022-10-30 09:05:12.289055: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit: UNKNOWN ERROR (303)
2022-10-30 09:05:12.289068: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host
(Fullstackdeveloper-desktop): /proc/driver/nvidia/version does not exist
2022-10-30 09:05:12.289215: I tensorflow/core/platform/cpu_feature_guard.cc:151] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2022-10-30 13:28:05.303259: W tensorflow/python/util/util.cc:368] Sets are not currently considered sequences, but this may change in the future, so consider
avoiding using them.
WARNING:absl:Found untraced functions such as head/dense_layer_call_fn, head/dense_layer_call_and_return_conditional_losses, re_lu_48_layer_call_fn,
re_lu_48_layer_call_and_return_conditional_losses while saving (showing 4 of 4). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: /media/fullstack-developer/0fab26ec-e67c-4a43-93fc-0ecfd4363f86/developer/workspace/big transfer/model_multi_weather_ssl/
assets
INFO:tensorflow:Assets written to: /media/fullstack-developer/0fab26ec-e67c-4a43-93fc-0ecfd4363f86/developer/workspace/big transfer/model_multi_weather_ssl/
assets
In [2]:

```

Figura 3.4 Proceso de preentrenamiento auto supervisado óptimo

Fine tuning supervisado con Weather Dataset sin preentrenamiento

Para poder demostrar la utilidad y mejora del modelo propuesto es necesario compararlo y evaluarlo con métricas de desempeño. Por lo que a continuación, definimos brevemente las métricas usadas.

Para la evaluación de los modelos haremos uso de 3 métricas de desempeño frecuentemente usadas para evaluar modelos de inteligencia artificial junto a la matriz de confusión.

- ✓ **Precisión:** es la calidad de la respuesta positiva del clasificador.

$$\frac{TP}{TP + FP}$$

- ✓ **Exhaustividad (Recall):** es la proximidad entre el resultado y la clasificación exacta.

$$\frac{TP}{TP + FN}$$

- ✓ **F1 Score:** Representa la relación entre las medidas de precisión y exhaustividad. Esto es de gran utilidad para comparar el rendimiento combinado de la precisión y la exhaustividad.

$$\frac{\text{Precisión} + \text{Exhaustividad}}{2}$$

Antes de continuar con el cálculo de la matriz de confusión es importante definir algunos términos:

Verdaderos positivos (TP – *true-positive*): Cuando una clase entrenada es reconocida correctamente por el algoritmo.

Falsos Positivos (FP – *false-positive*): Cuando una clase no entrenada es reconocida o confundida por el algoritmo por alguna clase que sí fue entrenado.

Falsos Negativos (FN – *false-negative*): Cuando el algoritmo no es capaz de reconocer a una clase que sí fue previamente entrenada.

Verdadero Negativo (TN – *true-negative*): Cuando una clase no entrenada no es reconocida por el algoritmo.

Matriz de confusión: contiene información acerca de las predicciones realizadas por un método o sistema de clasificación, comparado con el conjunto de clases en la tabla de aprendizaje o de *testing*. Esto es, comparar la predicción dada por el método versus la clase a la que estos realmente pertenecen. La Figura 3.5 muestra la distribución de la matriz de confusión.

| | | | | |
|---------------|-----------|------------|----|----|
| | | Predicción | | |
| | | Si | No | |
| Clases Reales | Positivas | TP | FN | =P |
| | Negativas | FP | TN | =N |

Figura 3.5 Matriz de confusión

El fine tuning supervisado sin preentrenamiento auto supervisado ha sido llevado a cabo con el objetivo de demostrar mediante evidencia que el enfoque propuesto en este trabajo de investigación tiene gran utilidad en el campo del reconocimiento del clima.

En la **¡Error! No se encuentra el origen de la referencia.**, se pueden observar los resultados obtenidos por el modelo sin preentrenamiento previo.

Tabla 2 Resultados obtenidos sin preentrenamiento

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| cloudy | 0.79 | 0.82 | 0.81 | 89 |
| rainy | 0.78 | 0.89 | 0.83 | 55 |
| shine | 0.31 | 0.14 | 0.19 | 36 |
| other | 0.69 | 0.75 | 0.72 | 100 |
| accuracy | | | 0.72 | 280 |
| macro avg | 0.64 | 0.65 | 0.64 | 280 |
| weighted avg | 0.69 | 0.72 | 0.70 | 280 |

Como se observa en la **¡Error! No se encuentra el origen de la referencia.**, los resultados obtenidos en estas pruebas no superaron el 72% por lo que claramente podemos concluir que con este modelo difícilmente se pueden alcanzar los objetivos planteados, a menos que enfoquemos los esfuerzos en optimizar los conjuntos de datos utilizados para el fine tuning supervisado.

Fine tuning supervisado Weather Dataset con preentrenamiento

La fase de fine tuning supervisado con preentrenamiento previo puede, en gran medida, mejorar el rendimiento de los modelos de inteligencia artificial. En el problema del reconocimiento automático del clima, verificamos que los resultados del modelo han mejorado en valores que van del 72% al 93%. En la Figura 3.6 se puede apreciar el proceso de entrenamiento con notable mejora.

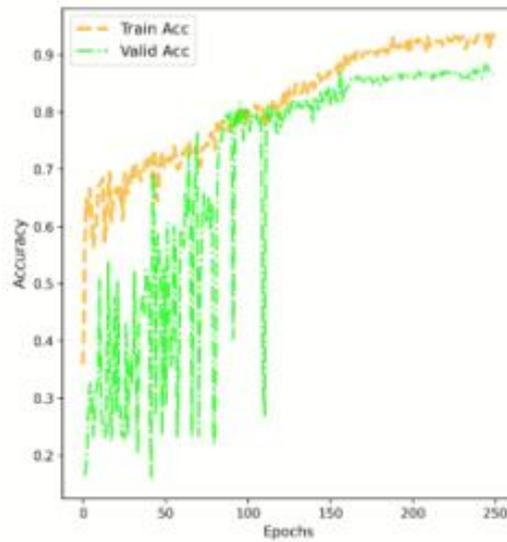


Figura 3.6 Proceso de fine tuning supervisado con preentrenamiento

En base al modelo entrenado, realizamos las respectivas pruebas de validación con el conjunto de datos previamente seleccionado. El conjunto de datos de validación está compuesto por 280 imágenes en total, 70 imágenes por cada clase.

En la **¡Error! No se encuentra el origen de la referencia.**, se puede observar la matriz de confusión obtenida de las pruebas realizadas.

Tabla 3 Matriz de confusión multiclase

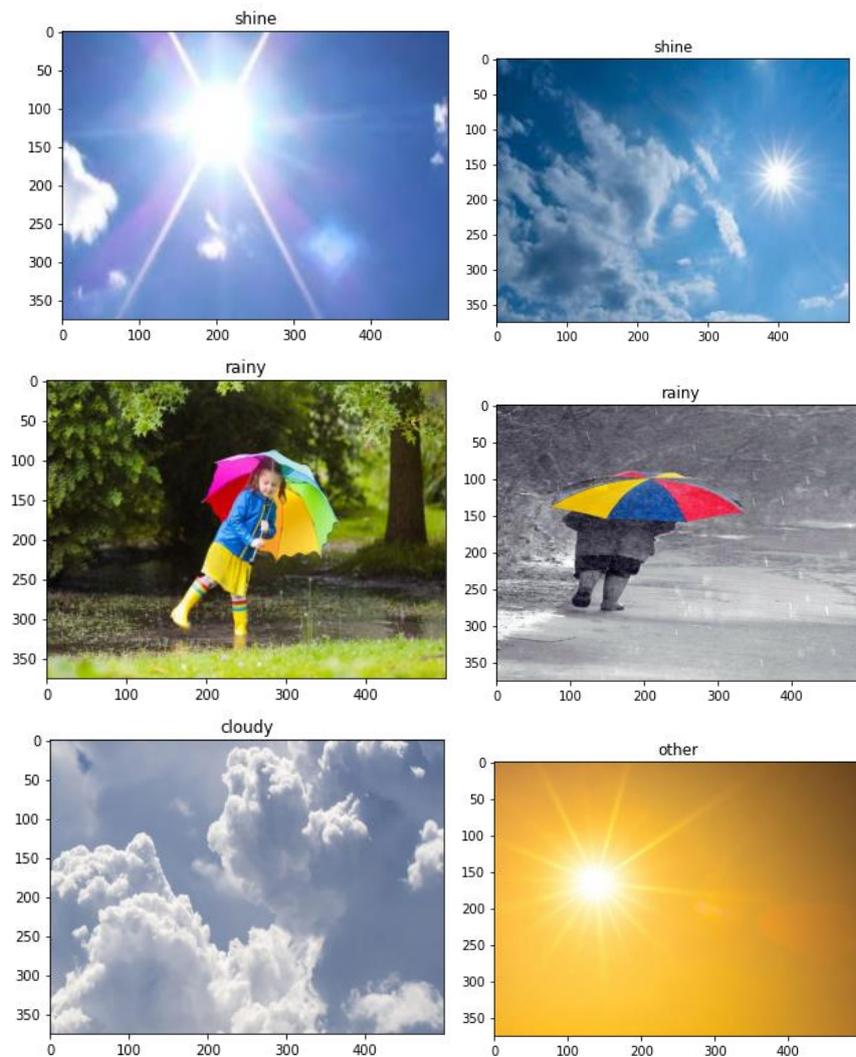
| | Cloudy | Rainy | Shine | Other | Total |
|--------|--------|-------|-------|-------|-------|
| Cloudy | 67 | 2 | 0 | 1 | 70 |
| Rainy | 3 | 65 | 0 | 2 | 70 |
| Shine | 2 | 2 | 63 | 3 | 70 |
| Other | 2 | 1 | 3 | 64 | 70 |
| Total | 74 | 70 | 66 | 70 | 280 |

Con los valores de la matriz de confusión se procede a calcular las métricas de desempeño previamente definidas. La **¡Error! No se encuentra el origen de la referencia.**, contiene los valores de precisión, recall y f1 score calculados.

Tabla 4 Resultados obtenidos con preentrenamiento

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| cloudy | 0.96 | 0.91 | 0.93 | 74 |
| rainy | 0.93 | 0.93 | 0.93 | 70 |
| shine | 0.90 | 0.85 | 0.93 | 66 |
| other | 0.91 | 0.91 | 0.91 | 70 |
| accuracy | | | 0.93 | 280 |
| macro avg | 0.93 | 0.90 | 0.93 | 280 |
| weighted avg | 0.93 | 0.90 | 0.93 | 280 |

A continuación, se muestra unos ejemplos de las predicciones realizadas por el modelo final.



3.2. Discusión

El modelo fue implementado en lenguaje de programación Python usando el framework de inteligencia artificial “TensorFlow” desarrollado por Google. Las pruebas fueron desarrolladas bajo diversas configuraciones de parámetros, como: épocas, batch size, learning rate y cantidad de imágenes de entrenamiento.

Una resnet-50, con 15 épocas de entrenamiento, batch size de 16, Learning rate inicial de 0,00009 que se ajusta en función del avance del entrenamiento a razón del 10% de reducción en cada ajuste, fue la mejor configuración, alcanzando una precisión promedio del 93%.

El método propuesto también fue evaluado con métricas de desempeño TOP-3, alcanzando una precisión promedio del 98% tal como se puede apreciar en la Figura 3.7.

```
In [4]: runcell(0, '/media/fullstack-developer/0fab26ec-e67c-4a43-93fc-0ecfd4363f86/
developer/workspace/big_transfer/test_dataset.py')
Total examples: 280
Total correct predictions: 275
Total wrong predictions: 5
Accuracy (%): 98.21428571428571
```

Figura 3.7 Métrica de desempeño TOP3

En la Tabla , hacemos una comparación entre los resultados obtenidos por los diferentes modelos de reconocimiento del clima propuestos en el estado del arte y el propuesto en este trabajo de investigación.

Tabla 5 Comparación métodos de reconocimiento del clima

| Método | Año | Imágenes etiquetadas | Precisión |
|---------------------------------------|-------------|----------------------|------------|
| aprendizaje multitarea [32] | 2017 | 10000 | 93,80% |
| Kernel múltiple [2] | 2016 | 20000 | 71% |
| Cámaras como sensores climáticos [19] | 2017 | 183798 | 90% |
| Extreme weather [4] | 2017 | 16635 | 94,50% |
| WeatherNet [3] | 2019 | 23865 | 93,80% |
| Método propuesto | 2023 | 1148 | 93% |

Si bien es cierto que el método propuesto en [4], ha alcanzado un mejor desempeño de detección del clima, una mejora notablemente importante en nuestro modelo es la cantidad de imágenes etiquetadas usadas para el entrenamiento **16635** en [4] vs **1148** en nuestro método propuesto.

4. CONCLUSIONES

En el estado del arte hemos encontrado varios trabajos que enfocan sus esfuerzos en el problema del reconocimiento automático del clima. Generalmente, estos trabajos hacen uso de técnicas tradicionales de aprendizaje profundo, por lo que gran parte de estos esfuerzos tienen que ser destinados a la construcción de conjuntos de datos etiquetados robustos que permitan un adecuado entrenamiento. En este trabajo de investigación, complementamos a los métodos tradicionales de aprendizaje profundo con técnicas de aprendizaje auto supervisado, fine tuning y transfer learning para el reconocimiento del clima en imágenes digitales.

El aprendizaje auto supervisado, presenta grandes ventajas en la construcción de modelos de reconocimiento del clima cuando se complementa correctamente con el resto de las técnicas tradicionales de aprendizaje de máquina, tal como: la transfer learning y el fine tuning. Por ello, es importante aclarar que el aprendizaje auto supervisado no es una nueva alternativa a las técnicas existentes, sino un complemento a las mismas que permite optimizar el uso de recursos en la construcción de un modelo de aprendizaje profundo.

La principal ventaja del uso del aprendizaje auto supervisado es la disminución considerable de la cantidad de imágenes etiquetadas requeridas para el entrenamiento supervisado, por lo que la necesidad de buenas capacidades computacionales para procesamiento gráfico aún es importante en estas condiciones.

La arquitectura implementada que alcanzó los mejores resultados es una red neuronal convolucional muy profunda, Resnet-50, con una fase de preentrenamiento supervisado con Imagenet, un preentrenamiento auto supervisado con Harvard Dataverse, y finalmente una fase de fine tuning supervisado con Weather Dataset. El modelo obtenido

de cada fase comparte sus variables de aprendizaje con la siguiente fase mediante transfer learning. El modelo fue entrenado en un servidor de 100GB de memoria física y procesador gráfico de 8GB.

Los resultados obtenidos son muy prometedores teniendo en cuenta que se ha entrenado con una pequeña cantidad de imágenes etiquetadas utilizadas en comparación con los trabajos relacionados. Sin embargo, es necesario reconocer que más recursos computacionales son fundamentales para entrenar un modelo mejor. Es muy importante planificar los requisitos de recursos computacionales cuando se entrena un modelo de inteligencia artificial con imágenes, ya que requiere de una alta disponibilidad de memoria física y procesamiento gráfico.

Es indispensable el uso de transfer learning para entornos donde se requiere optimizar el uso de recursos computacionales y el tiempo de implementación de la red. Gracias a esta técnica podemos hacer uso del modelo pre entrenado en Imagenet y de muchos otros modelos pre entrenados que requieren de mucho tiempo y recursos computacionales entrenarlos.

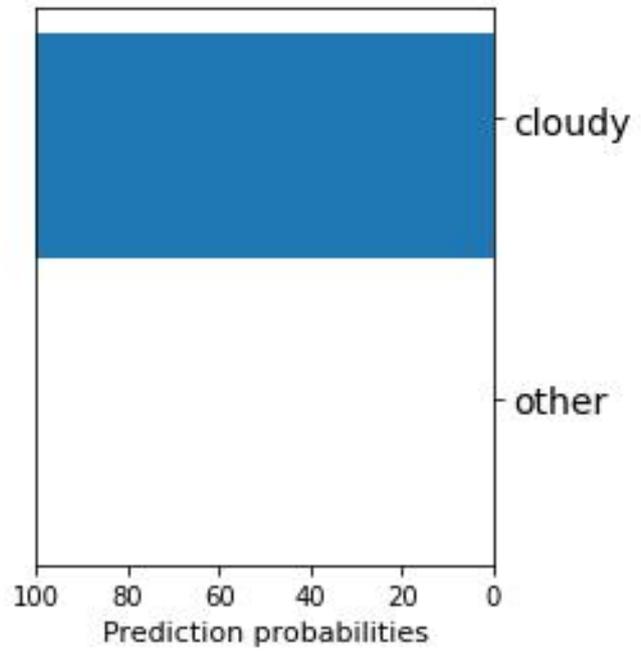
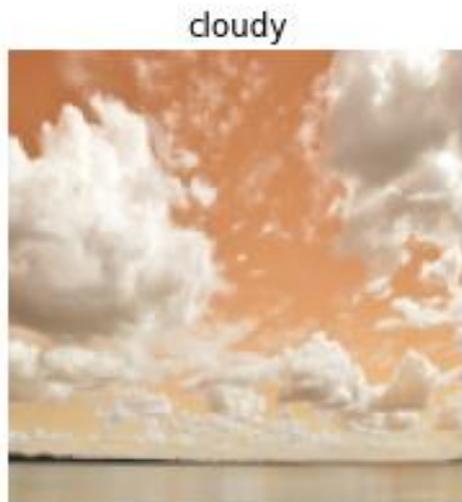
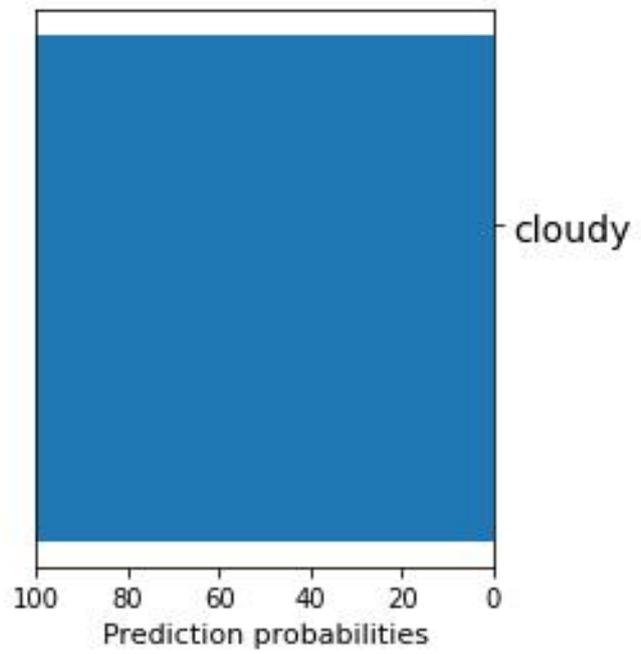
5. REFERENCIAS BIBLIOGRÁFICAS

- [1] “¿qué es el clima y cuál es su importancia?” <https://aleph.org.mx/que-es-el-clima-y-cual-es-su-importancia> (accessed Nov. 11, 2021).
- [2] Z. Zhang, H. Ma, H. Fu, and C. Zhang, “Scene-free multi-class weather classification on single images,” 2016, doi: 10.1016/j.neucom.2016.05.015.
- [3] M. R. Ibrahim, J. Haworth, and T. Cheng, “WeatherNet: Recognising Weather and Visual Conditions from Street-Level Images Using Deep Residual Learning,” *ISPRS International Journal of Geo-Information 2019, Vol. 8, Page 549*, vol. 8, no. 12, p. 549, Nov. 2019, doi: 10.3390/IJGI8120549.
- [4] Z. Zhu, L. Zhuo, P. Qu, K. Zhou, and J. Zhang, “Extreme weather recognition using convolutional neural networks,” *Proceedings - 2016 IEEE International Symposium on Multimedia, ISM 2016*, pp. 621–625, 2017, doi: 10.1109/ISM.2016.81.
- [5] J. Xu, Y. Pan, X. Pan, S. Hoi, Z. Yi, and Z. Xu, “RegNet: Self-Regulated Network for Image Classification,” pp. 1–6, 2021, [Online]. Available: <http://arxiv.org/abs/2101.00590>
- [6] P. Goyal *et al.*, “Self-supervised Pretraining of Visual Features in the Wild,” 2021.
- [7] A. Saeed, T. Ozcelebi, and J. Lukkien, “Multi-task Self-Supervised Learning for Human Activity Detection,” *Proc ACM Interact Mob Wearable Ubiquitous Technol*, vol. 3, no. 2, pp. 1–30, 2019, doi: 10.1145/3328932.
- [8] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, “Curiosity-driven exploration by self-supervised prediction,” *34th International Conference on Machine Learning, ICML 2017*, vol. 6, pp. 4261–4270, 2017.
- [9] J. Li, J. Liu, Y. Wong, S. Nishimura, and M. S. Kankanhalli, “Self-supervised representation learning using 360° data,” *MM 2019 - Proceedings of the 27th ACM International Conference on Multimedia*, pp. 998–1006, 2019, doi: 10.1145/3343031.3351019.
- [10] S. Azizi *et al.*, “Big Self-Supervised Models Advance Medical Image Classification,” Jan. 2021, Accessed: Nov. 13, 2021. [Online]. Available: <https://arxiv.org/abs/2101.05224v2>
- [11] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” *37th International Conference on*

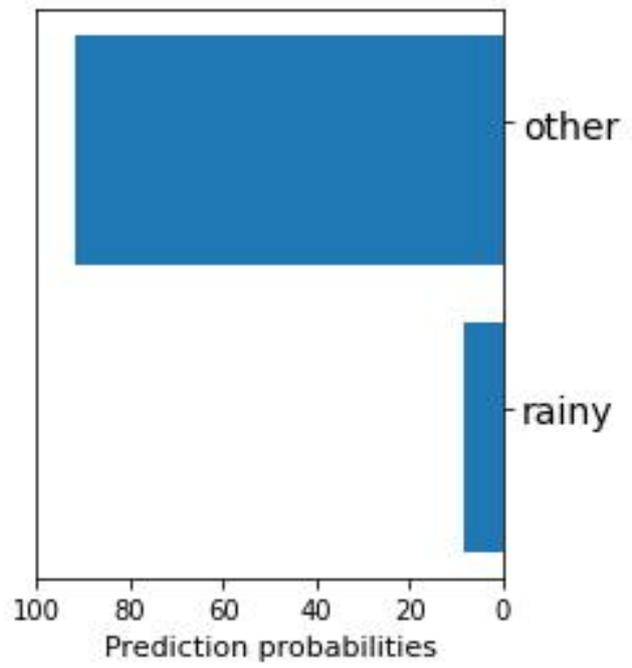
- Machine Learning, ICML 2020*, vol. PartF16814, no. Figure 1, pp. 1575–1585, 2020.
- [12] T. Gevers and A. Smeulders, “Identity Mappings in Deep Residual Networks,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9909 LNCS, p. V, 2016, doi: 10.1007/978-3-319-46493-0.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
- [14] “ImageNet.” <https://image-net.org/> (accessed Dec. 06, 2022).
- [15] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans Knowl Data Eng*, vol. 22, no. 10, pp. 1345–1359, 2010, doi: 10.1109/TKDE.2009.191.
- [16] “Machine learning (ML) is expanding the possibilities - LogPoint.” <https://www.logpoint.com/da/blog/explained-siemply-machine-learning/> (accessed Sep. 14, 2022).
- [17] “Evaluating models using the Top N accuracy metrics | by Rushabh Nagda | NanoNets | Medium.” <https://medium.com/nanonets/evaluating-models-using-the-top-n-accuracy-metrics-c0355b36f91b> (accessed Jan. 29, 2023).
- [18] X. Li, Z. Wang, and X. Lu, “A multi-task framework for weather recognition,” *MM 2017 - Proceedings of the 2017 ACM Multimedia Conference*, pp. 1318–1326, 2017, doi: 10.1145/3123266.3123382.
- [19] W. T. Chu, X. Y. Zheng, and D. S. Ding, “Camera as weather sensor: Estimating weather information from single images,” *J Vis Commun Image Represent*, vol. 46, pp. 233–249, 2017, doi: 10.1016/j.jvcir.2017.04.002.
- [20] J. Xu, Y. Pan, X. Pan, S. Hoi, Z. Yi, and Z. Xu, “RegNet: Self-Regulated Network for Image Classification,” pp. 1–6, 2021, [Online]. Available: <http://arxiv.org/abs/2101.00590>
- [21] “Road CCTV images with associated weather data - Harvard Dataverse.” <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/SV9N9F> (accessed Apr. 04, 2022).
- [22] A. Kolesnikov *et al.*, “Big Transfer (BiT): General Visual Representation Learning,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial*

- Intelligence and Lecture Notes in Bioinformatics*), vol. 12350 LNCS, pp. 491–507, Dec. 2019, doi: 10.48550/arxiv.1912.11370.
- [23] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *Int J Comput Vis*, vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/S11263-015-0816-Y.
- [24] D. Grabowski and A. Czyżewski, “System for monitoring road slippery based on CCTV cameras and convolutional neural networks,” *J Intell Inf Syst*, vol. 55, no. 3, pp. 521–534, Dec. 2020, doi: 10.1007/S10844-020-00618-5/FIGURES/14.
- [25] D. Grabowski and A. Czyżewski, “System for monitoring road slippery based on CCTV cameras and convolutional neural networks,” *J Intell Inf Syst*, vol. 55, no. 3, pp. 521–534, Dec. 2020, doi: 10.1007/S10844-020-00618-5/FIGURES/14.
- [26] “Weather dataset | Kaggle.” <https://www.kaggle.com/datasets/vijaygiitk/multiclass-weather-dataset> (accessed Apr. 04, 2022).
- [27] “Home Page - Scorecard Research.” <https://www.scorecardresearch.com/> (accessed Dec. 13, 2022).
- [28] “Beautiful Free Images & Pictures | Unsplash.” <https://unsplash.com/> (accessed Dec. 13, 2022).
- [29] “Stock Images, Photos, Vectors, Video, and Music | Shutterstock.” <https://www.shutterstock.com/> (accessed Dec. 13, 2022).
- [30] “CIFAR-10 and CIFAR-100 datasets.” <https://www.cs.toronto.edu/~kriz/cifar.html> (accessed Jan. 02, 2023).
- [31] X. Zhai *et al.*, “A Large-scale Study of Representation Learning with the Visual Task Adaptation Benchmark,” Oct. 2019, doi: 10.48550/arxiv.1910.04867.
- [32] X. Li, Z. Wang, and X. Lu, “A multi-task framework for weather recognition,” *MM 2017 - Proceedings of the 2017 ACM Multimedia Conference*, pp. 1318–1326, 2017, doi: 10.1145/3123266.3123382.

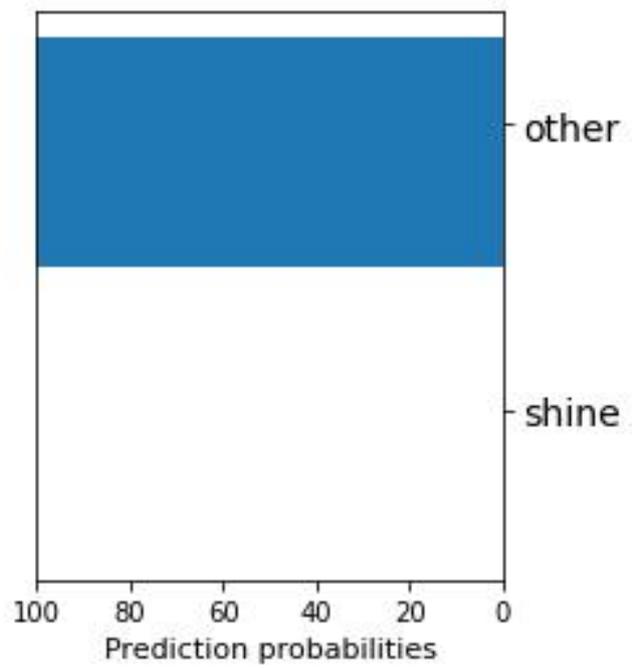
Anexo I – PROBABILIDADES DE PREDICCIÓN



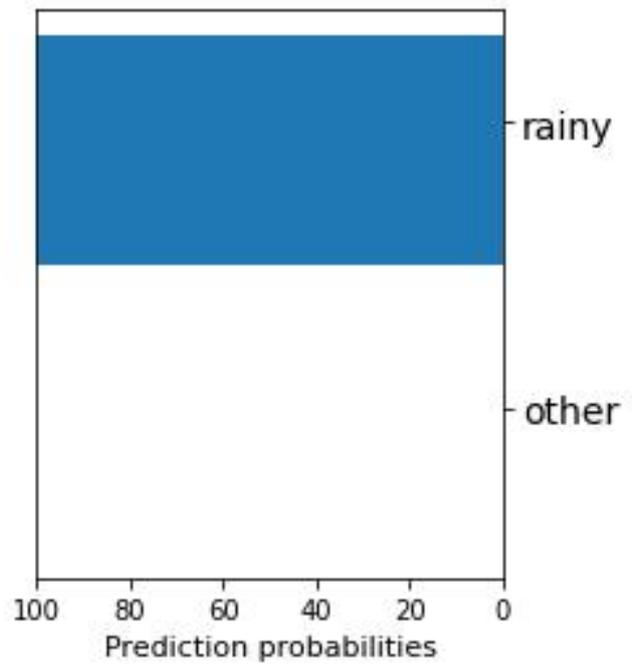
other



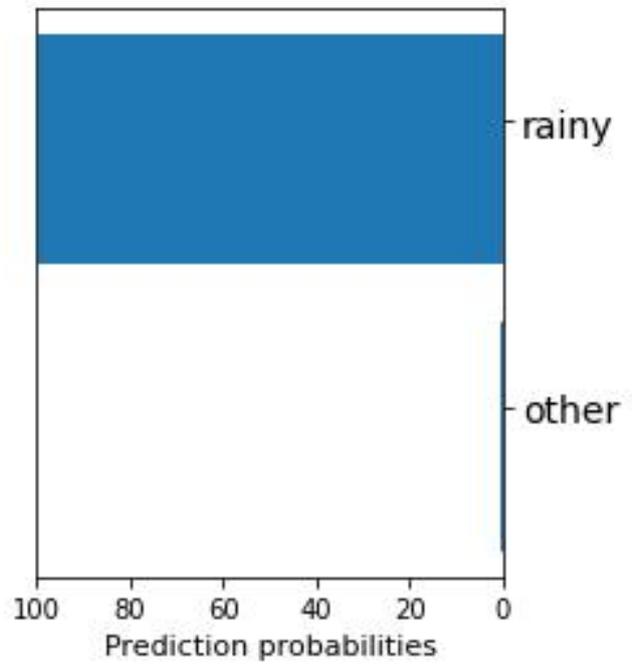
other



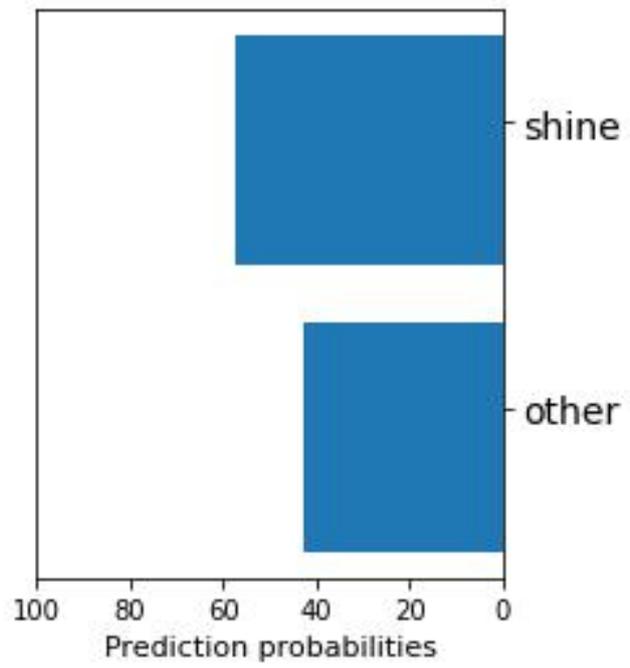
rainy



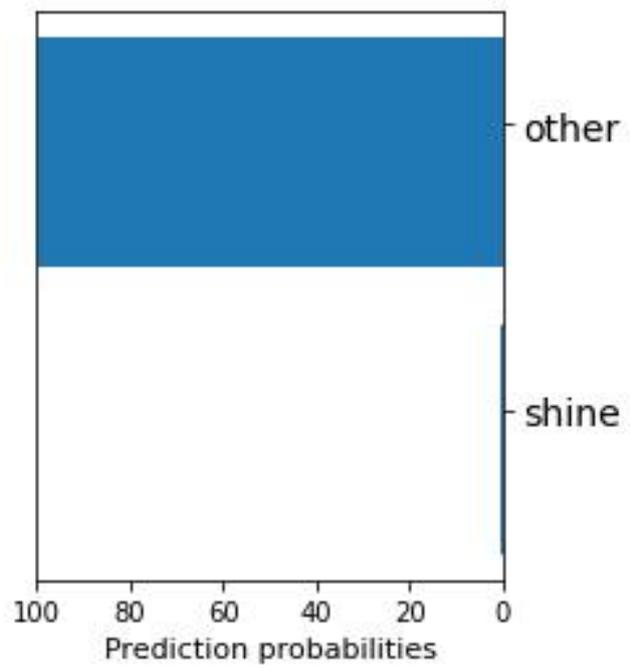
rainy

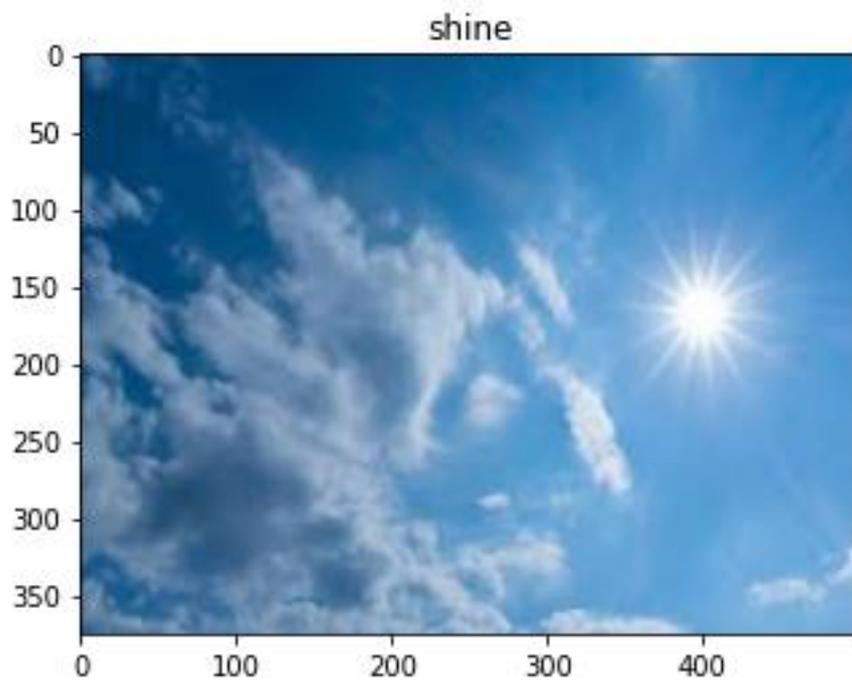
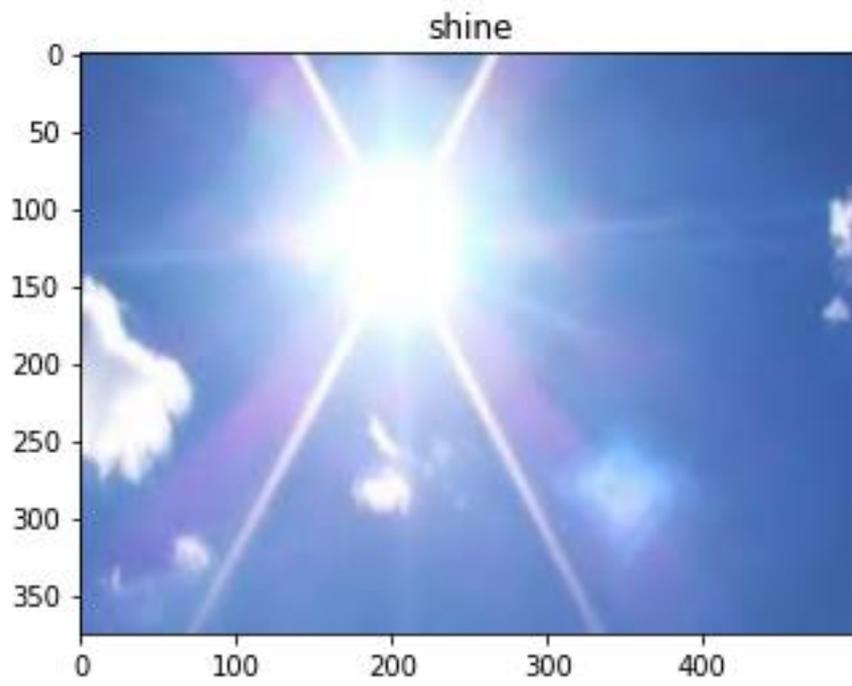


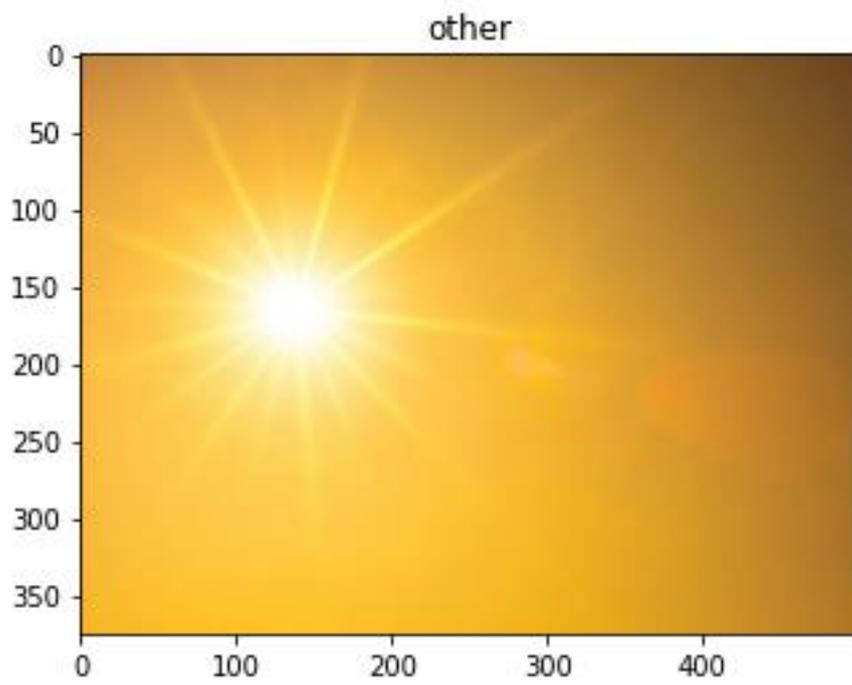
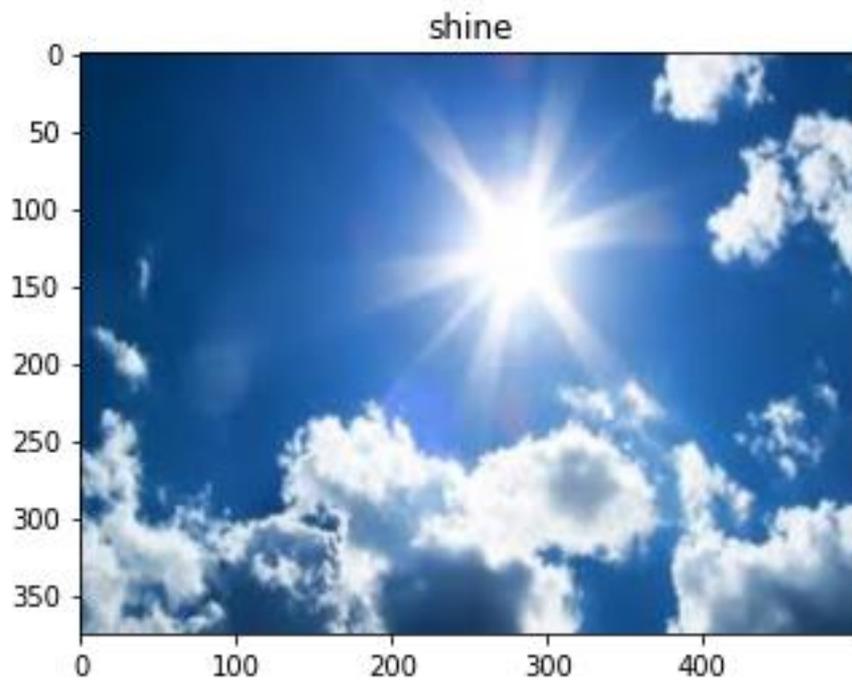
shine



shine





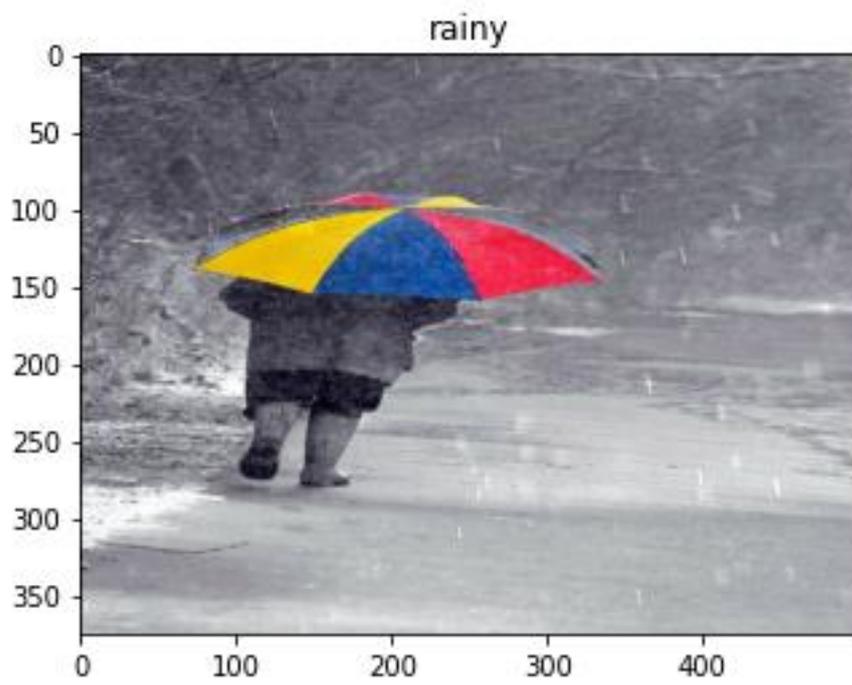
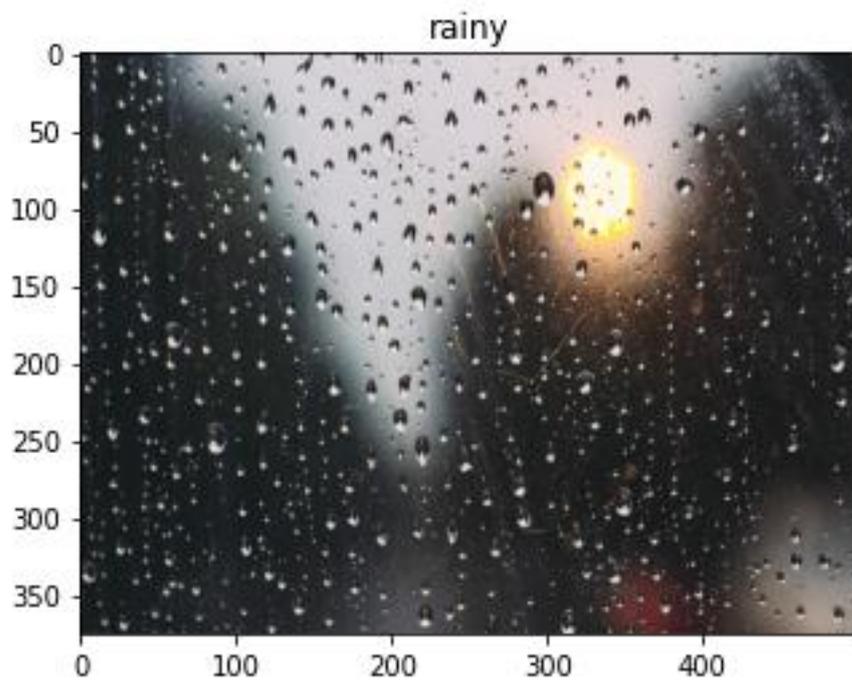


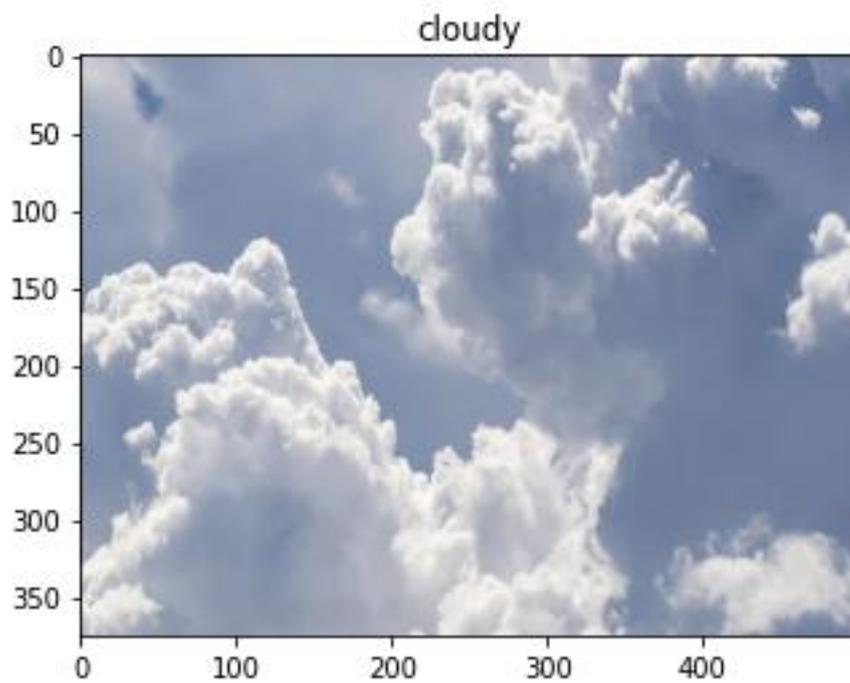
rainy



rainy







Anexo II – CÓDIGO FUENTE

El anexo II, consta de un CD con los archivos fuentes y ejecutables de todos los componentes del modelo. Así como también los conjuntos de datos configurados en tensorflow para cada una de las fases y los modelos generados.