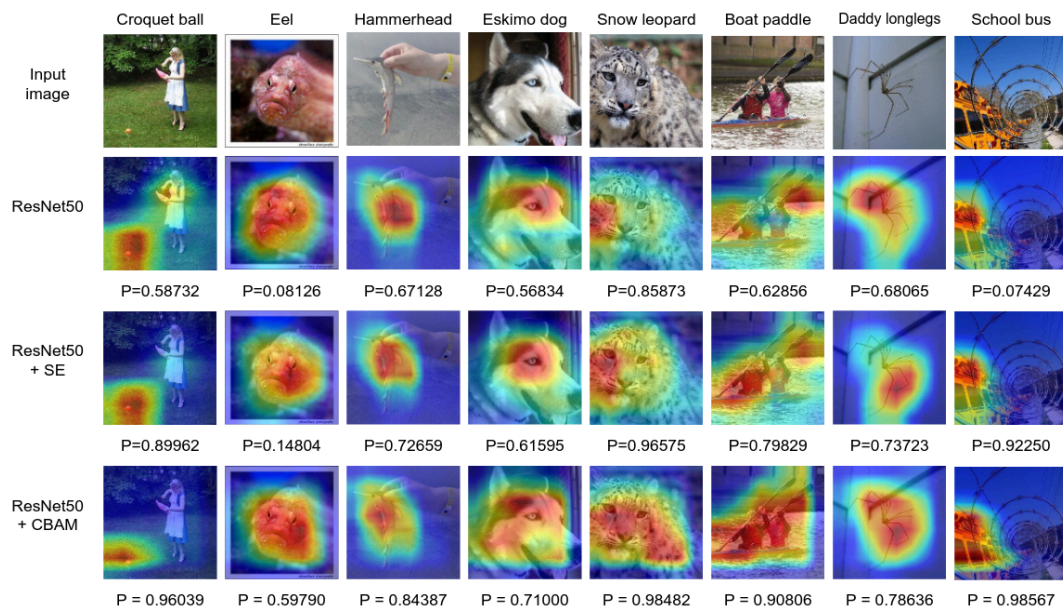


1. Clasificación de Imágenes



Visualizaciones de propagación de gradiente (Grad-CAM), Woo et al. "[CBAM: Convolutional Block Attention Module](#)". Este paper propone un mecanismo de atención entrenable y compara el desempeño al incluirlo a redes convolucionales conocidas.

La investigación académica tiende a enfocarse en aspectos técnicos novedosos y generalmente muestra resultados en conjuntos de datos de referencia estándar para permitir comparaciones justas con trabajos anteriores.

En el mundo real, los despliegues de visión por computadora a menudo modifican ambas suposiciones: tienden a utilizar ideas simples y comprobadas en lugar de modelos complejos de última generación y suelen utilizar conjuntos de datos personalizados adaptados al problema en cuestión. En muchas aplicaciones prácticas de visión por computadora, la tarea de construir y curar conjuntos de datos para entrenamiento y evaluación toma mucho más tiempo y esfuerzo que el proceso de construcción de los modelos.

De todas las ideas que hemos discutido en clase, la más útil en la práctica es la **clasificación de imágenes mediante aprendizaje por transferencia desde un modelo pre-entrenado**. Si en el futuro aplicas visión por computadora en tu trabajo, esta es probablemente la técnica que utilizarás. En este proyecto construirás un sistema de clasificación de imágenes tal como lo harías en una situación real.

Deberás recolectar tu propio conjunto de datos de clasificación de imágenes (puede ser un *dataset* preexistente de la web), dividirlo en particiones (entrenamiento / validación / prueba) y entrenar modelos de clasificación. Se les pide proponer una arquitectura convolucional diseñada por ustedes y comparar los resultados con un modelo obtenido mediante aprendizaje por transferencia desde redes preentrenadas en ImageNet.

Deberás explorar varias opciones y parámetros, y analizar el rendimiento de tus redes entrenadas. Además, existen técnicas avanzadas de visualización, interpretación y exploración de redes convolucionales, estas técnicas se las conoce como de “**Model Understanding**”, siendo el método **Grad-CAM** (Gradient-weighted Class Activation Mapping) uno de los más conocidos. Para este trabajo se les pide, entonces, desarrollar uno o más de estos métodos de interpretación y mostrar resultados sobre alguno de los modelos de clasificación obtenidos.

Links de información general sobre **Grad-CAM** y otros métodos de interpretación:

- En el artículo [Grad-CAM In PyTorch](#) podrán encontrar una introducción al método **Grad-CAM**
- La librería **Captum** para PyTorch está dedicada específicamente a esto: <https://pytorch.org/tutorials/beginner/introyt/captumyt.html>

Artículos recomendados

Además de lo visto en clase, estos son dos de los primeros artículos sobre aprendizaje por transferencia para redes convolucionales profundas:

- Donahue et al, “DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition”, ICML 2014. [\[arXiv\]](#)
- Razavian et al, “CNN Features off-the-shelf: an Astounding Baseline for Recognition”, CVPR Workshops 2014. [\[arXiv\]](#)

Al analizar un modelo de clasificación entrenado, un método de atribución intenta determinar qué porción de una imagen de entrada fue responsable de la decisión del modelo. Algunos artículos en esta área que podés leer son:

- Simonyan et al, “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”, ICLR Workshop 2014. [\[arXiv\]](#)
- Selvaraju et al, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”, IJCV 2019. [\[arXiv\]](#)
- Fong et al, “Understanding Deep Networks via Extremal Perturbations and Smooth Masks”, ICCV 2019. [\[arXiv\]](#)

Crear un Conjunto de Datos de Clasificación de Imágenes

La primera tarea en este proyecto es recolectar tu propio conjunto de datos de clasificación de imágenes para una tarea de clasificación que elijas.

Primero deberías definir un conjunto de dos o más categorías en las que quieras clasificar las imágenes. Estas categorías pueden ser amplias (p. ej., perro/gato/pez, hotdog/no-hotdog) o más específicas (por ejemplo, perro siberiano/perro salchicha).

¡Ponete creativo con las categorías que elijas!

Decidí si tus categorías son mutuamente excluyentes (una imagen tendrá exactamente una etiqueta correcta) o no excluyentes (una imagen puede tener cero o más etiquetas correctas). Las categorías mutuamente excluyentes son más simples de manejar y analizar, por lo que te sugerimos que intentes idear un conjunto de categorías mutuamente excluyentes.

Una vez que hayas decidido un conjunto de categorías, necesitarás construir un conjunto de datos recolectando y etiquetando imágenes. Los conjuntos de datos más grandes suelen dar mejores resultados, pero tu conjunto de datos debería tener al menos 100 imágenes por categoría.

[Google Dataset Search](#) es una herramienta moderna muy interesante que indexa conjuntos de datos de varios sitios (kaggle, roboflow, mendeley, figshare, etc). Ejemplo de búsqueda, [fruit classification](#).

Pueden explorar la librería [Selenium](#) para “scrapear” imágenes de la web, algunos links útiles: [How to Scrape Images from Google Using Python Selenium: End-to-End ML](#), [A simple Selenium image scrape from an interactive Google Image Search](#)

Usá la creatividad en la fuente de tus imágenes, pueden buscar imágenes en internet, mezclar datasets o armar tu propio conjunto de datos tomando fotos.

Luego de recolectar y etiquetar tu conjunto de datos, deberías dividirlo en conjuntos de entrenamiento, validación y prueba. Una regla general para dividir conjuntos de datos es asignar el 70% de tus datos a entrenamiento, 10% a validación y 20% a prueba.

Esperamos que el proceso de construir y etiquetar el conjunto de datos constituya una gran parte del trabajo para este proyecto.

En todos los casos el modelo final debe ser probado con imágenes nuevas no contenidas en el conjunto de datos.

Tu informe debería incluir una sección que describa tu tarea de clasificación y tu conjunto de datos. Explicá qué tarea de clasificación estás tratando de resolver, por qué podría ser útil y cuáles son las categorías. Describí tu proceso de recolección y etiquetado de imágenes, y discutí cualquier desafío que enfrentaste al construir tu conjunto de datos. Finalmente, mostrá ejemplos de tu conjunto de datos para dar una idea de los tipos de imágenes que contiene.

Entrenar un Clasificador de Imágenes

Una vez que hayas recolectado un conjunto de datos, es hora de entrenar varios modelos en este conjunto. Pueden proponer una arquitectura de red neuronal diseñada por ustedes mismos, pero no deben cuidar el tiempo que le dedican a esto, ¡el transfer learning es la norma no la excepción!

Deberán utilizar torchvision, que proporciona [una amplia variedad de modelos pre entrenados](#). Te recomendamos comparar múltiples arquitecturas, pero como primer paso te sugerimos usar ResNet18, ResNet50 o un modelo pequeño de RegNet (p. ej., RegNetY-400MF).

Podés considerar tres maneras diferentes de entrenar un clasificador:

- Entrenar desde cero: Inicializar el modelo aleatoriamente y entrenarlo desde cero en tu conjunto de entrenamiento.
- Extracción de características: Inicializar el modelo usando pesos pre entrenados en ImageNet; extraer características del modelo y entrenar un modelo lineal sobre las características extraídas. Esto puede implementarse de varias maneras: podés usar PyTorch para agregar una nueva capa al modelo, congelar todas menos la última capa y entrenar en PyTorch; también podés extraer características del modelo y guardarlas en algún formato, luego entrenar un clasificador lineal con otro paquete como scikit-learn o una librería de boosting.
- Ajuste fino (Fine-Tuning): Inicializar el modelo usando pesos pre entrenados en ImageNet; y ajustar una porción (o todo) el modelo en tu conjunto de entrenamiento. Esto debería funcionar al menos tan bien como la extracción de características.

Es posible que necesites experimentar con diferentes optimizadores, tasas de aprendizaje, regularización L2 y estrategias de aumento de datos.

Deberán entrenar al menos 6 modelos diferentes y reportar su precisión tanto en los conjuntos de entrenamiento como de validación.

Deberán mostrar las curvas de entrenamiento para tus modelos, graficando la pérdida de entrenamiento por iteración, así como la precisión del modelo en los conjuntos de entrenamiento y validación en cada época (pasada por el conjunto de entrenamiento).

Podés intentar usar aumento en tiempo de prueba o ensambles de modelos para mejorar el rendimiento de clasificación de tus modelos individuales.

El informe deberá explicar los modelos que elegiste, describir los experimentos que realizaste y mostrar los resultados descritos arriba.

Analizar tus Modelos

Después de haber entrenado tus modelos, deberías analizarlos para intentar obtener información sobre qué tan bien están funcionando y en qué situaciones fallan. Deberías analizar al menos dos modelos diferentes: el modelo con mejor rendimiento y al menos otro modelo de tu elección.

Deberías mostrar lo siguiente:

- Ejemplos cualitativos de imágenes que fueron clasificadas correctamente e incorrectamente. Podés mostrar tanto imágenes seleccionadas manualmente para resaltar características interesantes de tu modelo, como imágenes seleccionadas al azar para dar una mejor idea del rendimiento promedio de tu modelo.
- Matriz de confusión: Una matriz 2D de categoría predicha versus categoría real, donde cada entrada muestra la fracción de imágenes del conjunto de validación que caen en esa situación; esto muestra los tipos de errores que comete tu modelo.
- Métodos de atribución: Usar uno o más métodos de atribución para dar ejemplos de qué porciones de las imágenes utiliza el modelo para tomar decisiones de clasificación.
- Graficar las características visuales extraídas por la red en las capas previas a la clasificación. Pueden utilizar métodos como reducción de la dimensionalidad PCA, o la técnica de t-SNE (t-distributed stochastic neighbor embedding, [Mastering t-SNE](#)).

El informe debería incluir secciones que describan y muestren cada uno de los métodos de análisis anteriores, así como cualquier otro tipo de análisis que realices.

Evaluación en el Conjunto de Prueba

Después de todo tu análisis, deberías correr tu mejor modelo en el conjunto de prueba. Si tu modelo tiene un rendimiento muy diferente en los conjuntos de validación y prueba, deberías intentar explicar por qué podría ser ese el caso.

Entregables

En resumen, en este proyecto esperamos que:

- Crees tu propio conjunto de datos de clasificación de imágenes con al menos 100 imágenes por categoría. En tu informe, discutí el problema de clasificación que elegiste resolver y cómo recolectaste el conjunto de datos. Mostrá imágenes de ejemplo del conjunto.
- Entrená al menos 6 modelos de clasificación de imágenes en tu conjunto de datos; en tu informe, mostrá curvas de entrenamiento para cada modelo (pérdida y precisión), así como una comparación de precisión en el conjunto de validación.
- Analizá el rendimiento de tus modelos mostrando ejemplos de clasificación correcta/incorrecta, matrices de confusión y métodos de atribución.
- Probá tu mejor modelo en el conjunto de prueba y discutí los resultados.