

Reporte Actividad Integradora 3



**Tecnológico
de Monterrey**

Sistemas de Visión Por Computadora

Francisco Ismael Sainz Williams

Marcelo Alejandro Salazar Martínez

Adrián Marcelo Suarez Ponce

Arturo Vázquez Muñoz

Para el desarrollo de este proyecto, fue necesario dividirlo en varias etapas para obtener el resultado más óptimo y el modelo con mayor precisión posible. Estas etapas son las de recolección y procesamiento de datos, selección y entrenamiento de modelos y, por último, la evaluación y métricas de desempeño. Todo esto nos ayudó a enfocarnos en lo más importante de cada etapa y asegurar que esté funcionando de manera correcta para mitigar errores y poder pasar al siguiente paso de manera efectiva. Al cumplirse todas las etapas se obtiene un modelo entrenado para recolección y clasificación de imágenes con distintas precisiones, dependiendo de los parámetros y la versión utilizada.

Primeramente, nos enfocamos en la recolección de datos que, en este caso, son imágenes obtenidas del sitio web shutterstock, por medio de la técnica de Web Scraping. Esta técnica nos sirve para obtener información de manera automatizada a través del internet, en donde se por medio de drivers que dependen del buscador que utilicemos, en este caso mozilla firefox, y el URL del sitio web, se hace la conexión y se obtiene la información. El código que se programo nos solicita como input algun parametro que deseemos buscar, así como la cantidad de imágenes que se busca obtener. Al utilizar un sitio web buscador de imágenes, se introduce el parámetro y nuestro obtiene cierta cantidad de imágenes que pertenezcan a la categoría establecida.

Selección y entrenamiento del modelo

Para la selección del modelo, utilizamos una arquitectura que estuviera basada en una red neuronal convolucional, que se especializa en la detección de patrones y nos puede ser útil para la recolección de imágenes. Una vez identificado el tipo de arquitectura, decidimos utilizar la técnica de transfer learning, que consiste en utilizar como base un modelo ya entrenado para ahorrar el costo computacional que conlleva el entrenamiento de un modelo a gran escala. Una vez descargado el modelo VGG16, basado en una red neuronal convolucional de 16 capas, decidimos importar todas sus capas a nuestro nuevo modelo y reemplazamos la capa de salida por una nueva con solo 5 neuronas, cada una correspondiendo al objeto que deseamos clasificar. Además utilizamos una función de activación softmax, que nos permite obtener un vector de probabilidades para cada objeto.

Para el entrenamiento de los modelos, decidimos experimentar con varias técnicas de fine tuning para ver el comportamiento del mismo set de datos con diferentes modelos. La técnica más simple, que repetimos dos veces, fue hacer freeze a todas las capas hasta la penúltima, para mantener los pesos ya entrenados y permitir que se entrene solamente la última capa. En un modelo decidimos hacer el entrenamiento con 5 epochs y en otro con 8 epochs, para ver la diferencia de rendimiento entre ambos y experimentar cómo aumentar la complejidad del modelo afecta las predicciones realizadas.

En las métricas de entrenamiento, pudimos observar que agregar más epochs no aumenta mucho el rendimiento de nuestro modelo, ya que este es lo suficientemente complejo como para poder realizar predicciones arriba del 70%. Los resultados del test de prueba arrojaron una precisión de ~80% en la mayoría de nuestras experimentaciones con diferentes modelos. Debido a esto, decidimos analizar las imágenes que fueron descargadas por nuestro web scraper y pudimos observar que existen imágenes clasificadas de manera errónea, debido a que fueron descargadas.

Finalmente, algo que pudimos identificar en nuestras métricas fue que, al momento de permitir el reentrenamiento de algunas de las capas aumenta la complejidad del modelo, haciendo que este mismo haga overfitting.