Reporte final de proyecto

Diego Isla López

Abstract: El problema de reconocimiento de objetos y formas es uno de los más estudiados tanto en aprendizaje automático como en visión computacional. Se hace un experimento con imágenes de Pokémon para verificar el desempeño de modelos tradicionales de clasificación así como de redes neuronales y hacer una comparación entre ellos.

I. Introducción

El problema de clasificación y reconocimiento de objetos es un problema clásico en Aprendizaje Automático (AA). En este proyecto se busca probar el rendimiento de métodos clásicos de clasificación y redes neuronales en el reconocimiento de diferentes pokemones. Si bien existe un gran número de pokemones distintos, nos enfocaremos en la identificación de un conjunto de pokemones muy similares, los llamados *eevolution*. Los *eevolution* son pokemones que surgen de la evolución de un mismo pokémon, Eevee, por lo que sus formas son muy parecidas (similares a perros o gatos) y con diferencias específicas en color y forma de alguna parte del cuerpo (figura 2). En particular, se intentará hacer la prueba de desempeño usando la dinámica de *Who's that Pokémon?* presente en la serie animada de Pokémon. La dinámica consiste en que el televidente adivine qué Pokémon se está mostrando a partir de una "sombra" donde la figura del pokemon se presenta en un color sólido y la silueta es la única información usada para adivinar (figura 1). Existen ocho diferentes *eevolution*: Vaporeon, Jolteon, Flareon, Espeon, Glaceon, Umbreon, Sylveon, Leafeon. Las clases serán representadas con los nombres de dichos pokemon.

(a) Pregunta

(b) Respuesta

Figura 1: Dinámica de *Who's that Pokémon?*

Figura 2: Los diferentes *eevolution*

De este modo, se utilizará un conjunto de entrenamiento con imágenes a color de cada pokémon y un conjunto de prueba con imágenes en color sólido. Las imágenes a color fueron obtenidas de un conjunto disponible en Kaggle[1] y mediante *image scraping* usando la búsqueda de Google. Las imágenes sólidas fueron creadas a partir de imágenes transparentes obtenidas de búsquedas en Google, las cuales fueron rellenadas con color negro.

II. Estado del arte

No es difícil imaginar las aplicaciones que puede tener la detección de objetos, desde el área médica hasta seguridad. Dada la popularidad que han ganado las redes neuronales en los últimos años, se han llevado a cabo diversos experimentos además de desarrollo de técnicas nuevas para el problema de clasificación de imágenes. Las redes neuronales convolucionales (CNN) han resultado ser de las más eficientes para abordar este tipo de problemas. Debido a esto, se han propuesto variaciones a este modelo de redes neuronales. Como se menciona en [5], las ventajas que tienen las CNN sobre los métodos tradicionales de clasificación son:

[1]Disponible en https://www.kaggle.com/thedagger/pokemon-generation-one

- Exhiben una representación jerárquica de características
- Poseen una capacidad expresiva mayor
- Ofrecen la oportunidad de optimizar diferentes aspectos de manera simultanea
- Hacen posible trasladar ciertos problemas de visión computacional a problemas de transformación de datos a dimensiones mayores.

Recientemente han aparecido modelos como las CNN basadas en regiones (R-CNN), las cuales funcionan determinando un número fijo de regiones de la imagen a evaluar [1]; el modelo *You Only Look Once* (YOLO) [3] el cual ha tenido resultados significativos en el reconocimiento de imágenes en video en tiempo real.

De igual manera, el modelo MobileNets fue propuesto por ingenieros de Google en [2]. Este modelo ha mostrado buen desempeño en problemas varios de visión computacional además de clasificación de imágenes. En los experimentos de este proyecto se utiliza un modelo basado en MobileNets para la implementación con redes neuronales.

Una técnica que ha dado buenos resultados es la de aprendizaje por transferencia (TF) [4], que se refiere a pre-entrenar una red para obtener ciertas características y estás alimentarlas a otra red para poder tener un mejor desempeño de aprendizaje.

III. Metodología

Al ser un problema de clasificación de imágenes es necesario realizar un proceso de extracción de características. Este proceso se refiere a realizar un pre-procesamiento de las imágenes donde se traduce la información de los píxeles y se crea una matriz de características. Esto ayuda a reducir la dimensión y complejidad de las imágenes. Debido a que el conjunto de datos recabado resultó con una cantidad baja de imágenes, se utilizaron técnicas de incremento de datos para tener un conjunto de 2000 imágenes por clase. Para procesar las imágenes aumentadas se utilizaron métodos de ruido, rotaciones e inversiones aleatorias.

III-A. Métodos tradicionales

Se hará una comparación entre distintos clasificadores: regresión logística (LR), K-vecinos más cercanos (KNN), bosques aleatorios (RF), descenso por gradiente estocástico (SGD) y máquinas de vectores de soporte (SVM). Para la extracción de características se utilizarán tres métodos:

- Momentos de imagen
- Texturas Haralick
- Histograma de colores
- Descriptor KAZE

III-B. Redes neuronales convolucionales

Se elige un modelo pre-entrenado de Tensorflow[2] basado en la arquitectura Mobilenet. Este modelo admite una opción para entrenarse con un conjunto de datos personalizado, por lo que se hacen pruebas con ambas modalidades del modelo.

[2]Disponible en https://tfhub.dev/google/tf2-preview/mobilenet_v2/feature_vector/4

IV. Experimentación y resultados

IV-A. Métodos tradicionales

En la figura 3 es posible ver que el clasificador RF es el que presenta mejor desempeño entre los modelos tradicionales. En la tabla I observamos que la precisión para el modelo RF es de 73.48 % con el conjunto de validación. Sin embargo, al probar el desempeño del modelo con las imágenes de prueba, se observa que el comportamiento no es ideal. Las predicciones apuntan hacia una sola clase para todas las imágenes del conjunto.

Como se puede ver en la figura 4 , la matriz de confusión para el modelo muestra la distribución de las predicciones sobre una sola clase.

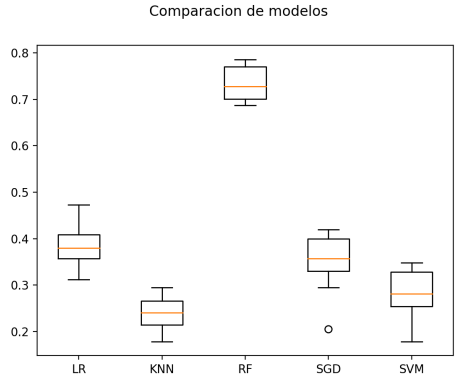


Figura 3: Desempeño de diferentes clasificadores

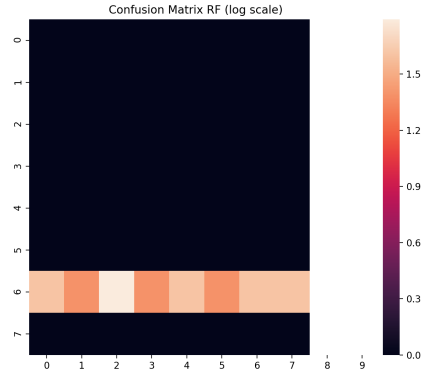


Figura 4: Matriz de confusión para RF

Modelo	Valor de pérdida	Precisión
LR	0.045492	38.21 %
KNN	0.036158	24.1 %
RF	0.03594	73.48 %
SGD	0.062634	35.08 %
SVM	0.048544	28.39 %

Tabla I: Tabla de resultados para CNN

IV-B. Redes neuronales convolucionales

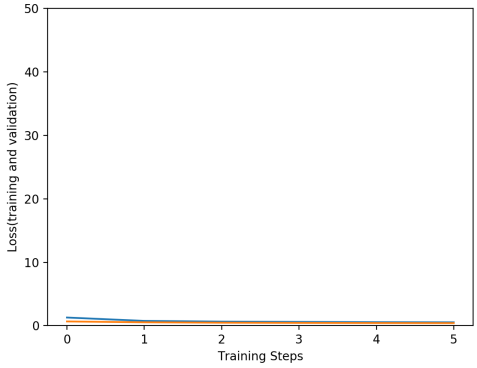
En la figura 5, correspondiente a la prueba con el modelo pre-entrenado, es posible ver que el valor de pérdida desde el

inicio no es alto y disminuye gradualmente con cada época del entrenamiento hasta llegar a un valor muy cercano a 0. Observando la gráfica de la evolución de la precisión, es notoria la diferencia entre el desempeño con las imágenes de entrenamiento (azul) y el desempeño entre las imágenes de validación (naranja). Asimismo, en la figura 6, correspondiente al rendimiento del modelo en modo de entrenamiento, es posible ver que la precisión con el conjunto de entrenamiento es muy cercana al 100 % y la precisión en la validación queda por debajo, alrededor de 85 %. Esto nos indica que está existiendo un sobreajuste hacia el conjunto de entrenamiento.

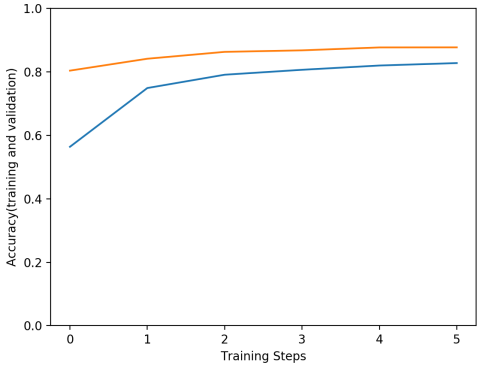
Conjunto	Valor de pérdida	Precisión
Validación	0.40	88.02 %
Prueba	2.02	36.67 %

Tabla II: Tabla de resultados para CNN

En la matriz de confusión para los resultados del modelo (figura 8) es posible ver la distribución de los resultados predichos. Es posible apreciar que existen dos clases que nunca aparecieron en las predicciones. Las clases con mayor cantidad de aciertos fueron la 5 y la 7 (mientras más claro el color, mayor frecuencia). Finalmente en la figura 7 podemos ver de manera gráfica el desempeño del modelo. Como se puede observar los resultados de la prueba están dentro del rango de precisión reportado en la tabla II.

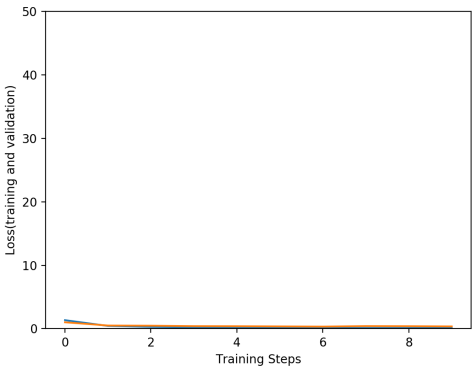


(a) Valor de pérdida

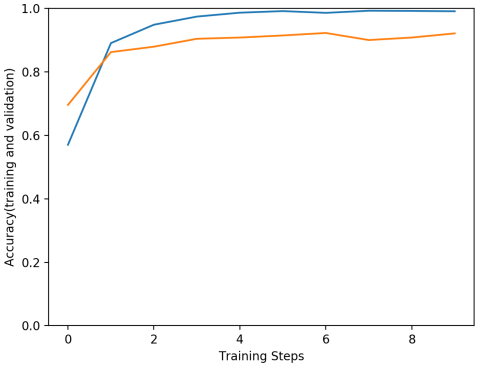


(b) Precisión de entrenamiento

Figura 5: Desempeño con 2000 datos de modelo pre-entrenado



(a) Valor de pérdida



(b) Precisión de entrenamiento

Figura 6: Desempeño con 2000 datos de modelo en modo entrenamiento

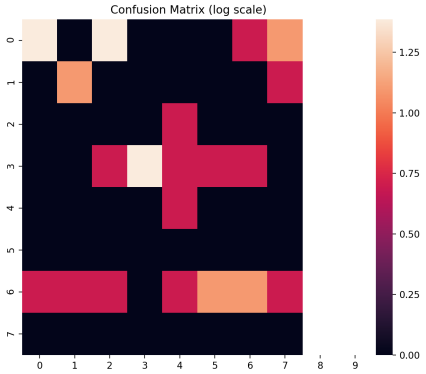
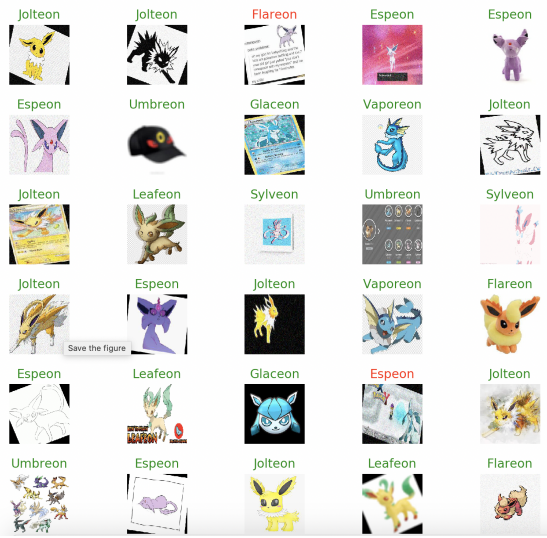


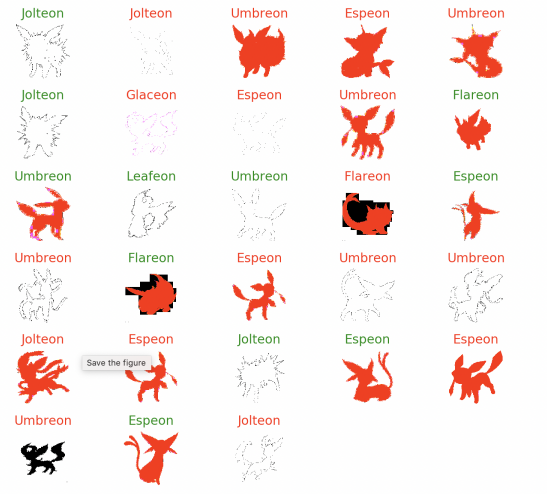
Figura 8: Matriz de confusión para CNN

V. Conclusiones

La diferencia de desempeño entre el modelo de CNN y los clasificadores tradicionales es clara. Es posible que la extracción de características de manera manual sea la causa de la falla en los modelos tradicionales, pues las imágenes de prueba al no tener características de color o texturas, este tipo de método de extracción podría no ser ideal. En el caso del modelo de CNN vemos que logra tener un desempeño



(a) Resultados de validación



(b) Resultados de predicción

Figura 7: Comparación de rendimiento

aceptable. Sin embargo, es recomendable explorar maneras de alcanzar un mejor desempeño, como utilizar técnicas de aprendizaje por transferencia. La calidad del conjunto de datos podría ser un aspecto a considerar en el desempeño del modelo CNN y los métodos tradicionales; si bien se utilizó un proceso de incremento de datos para lograr tener un conjunto de tamaño significativo, tener un conjunto considerable de imágenes diferentes podría tener un efecto positivo en el desempeño de los modelos. En el caso del modelo de CNN una manera de buscar una mejora en el desempeño sería probar con diferentes modelos de optimización, además de las mejoras en el conjunto de datos ya descritas.

Referencias

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[2] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[3] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2015.

[4] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks?, 2014.

[5] Zhong-Qiu Zhao, Peng Zheng, Shou tao Xu, and Xindong Wu. Object detection with deep learning: A review, 2018.