Music CNN

Abril, 2020
Machine Learning Models

Juan Diego Sique Martínez
Universidad Francisco Marroquín

Clasificación de elementos de notación musical

La notación musical es un sistema gráfico usado para representar el sonido en todos sus grados musicales, las modificaciones de tiempo, intensidad, articulación y matices que le afectan así como las pautas rítmicas y silencios que limitan su intervención.

Redes neuronales para la clasificación musical

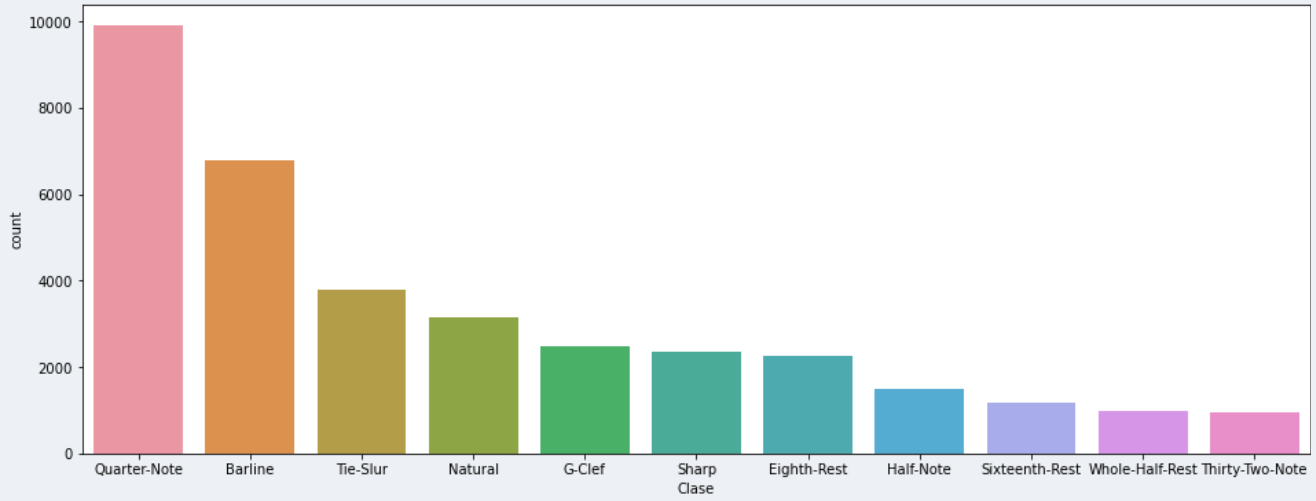
El proyecto consiste en obtener distintas imágenes de diversas fuentes para entrenar un modelo predictivo que clasifique imágenes según el símbolo de la notación musical que representan.
El modelo predictivo consistirá en una red neuronal convolucional, que mediante distintas técnicas predecirá el símbolo del que se trata según el gráfico de entrada.

¿Para qué puede utilizarse?

En el mercado existen muy pocos, por no decir uno o dos, software de transcripción de partituras hechas a mano para el ordenador. Con un modelo de precisión alta, podría intentarse desarrollar dicho software.

Datos y su preprocesamiento

Los datos fueron obtenidos de https://apacha.github.io/OMR-Datasets/, que consisten en un gran conjunto de imágenes. Luego con la librería CV2 de Python fueron transformados a un arreglo de bits que representa el contenido de la imagen.
Originalmente se escalaron las imágenes para un tamaño de 96^x96, pero por limitaciones físicas de mi computadora, se les cambió el tamaño a su tercera parte, es decir, 24^x24.
Son alrededor de 59 etiquetas, que fueron filtradas por importancia, eliminando las indicaciones de compás reduciendo a 30 clases diferentes.
Debido a que se consideró que eran demasiadas etiquetas se hizo un conteo con las clases más frecuentes y se eligieron las primeras 11, es decir la tercera parte de etiquetas.

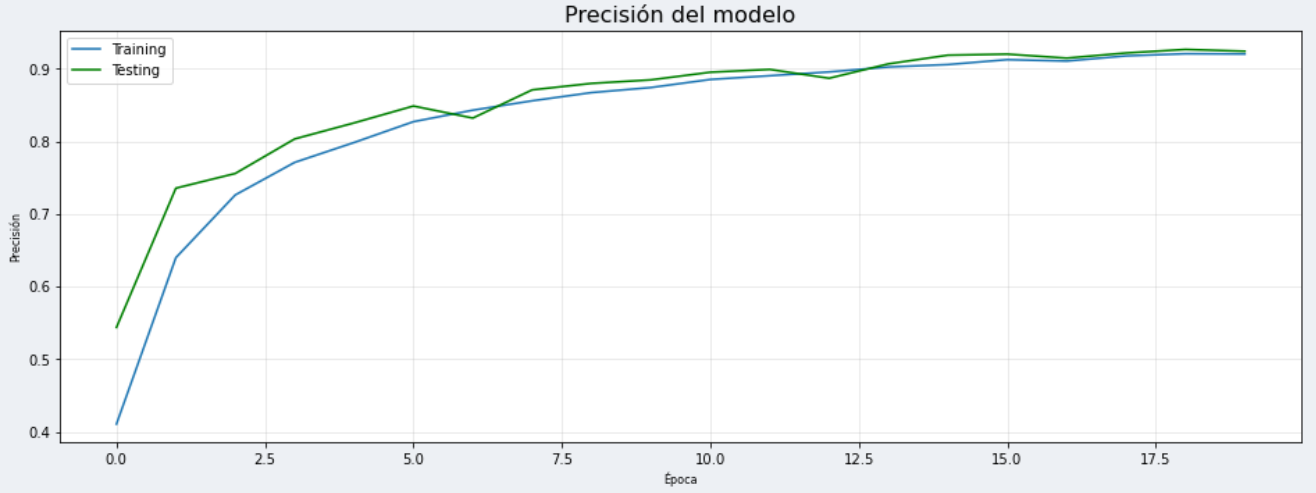


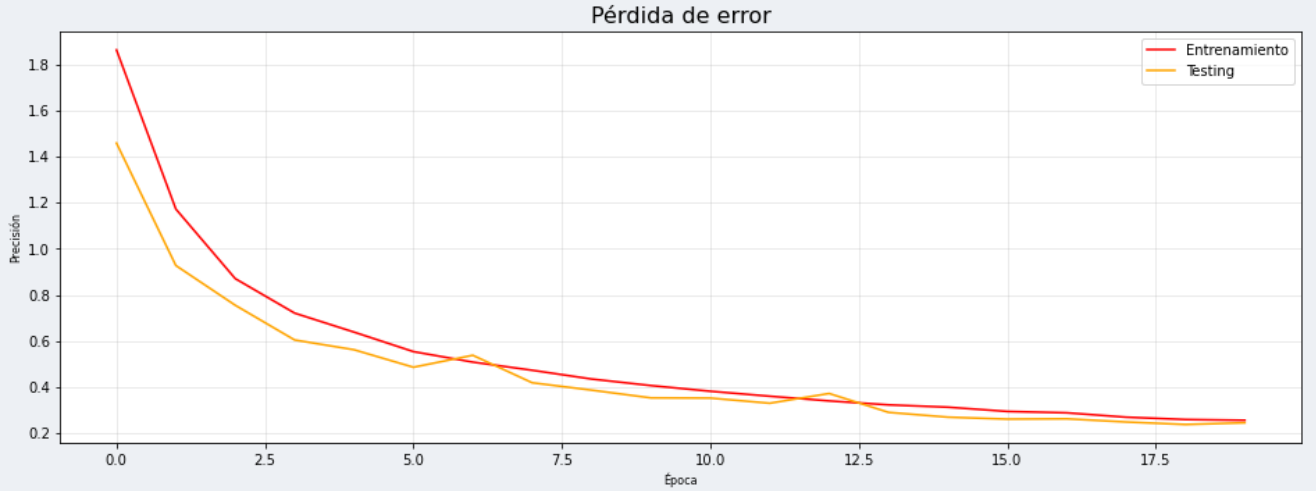
Una vez terminada la selección de clases se continúo el trabajo de entrenamiento con aproximadamente 29,000 elementos.

Modelo predictivo

El modelo consistía con dos capas convolucionales con filtros de 3*3 y 2*2, respectivamente. Se aplicó también *pooling* y un *dropout* del 25% de los datos para la primera capa. Luego se aplicó una densa usando como función de activación ReLu.

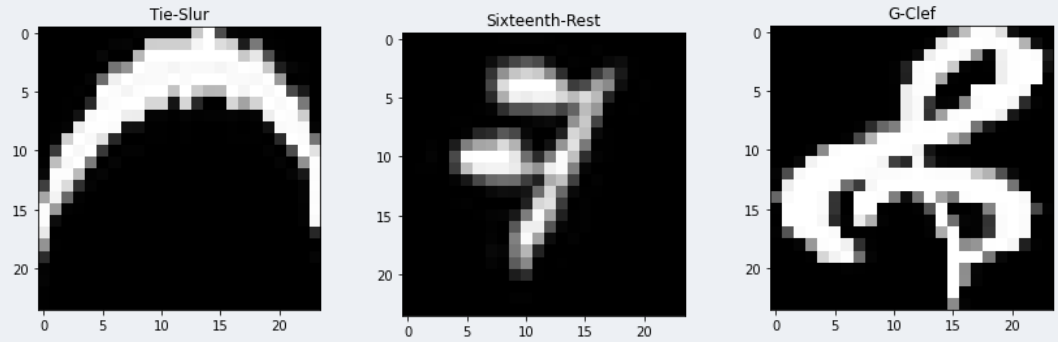
Tras entrenar el modelo con 20 épocas se pudo percatar de la mejoría en la precisión durante el entrenamiento. Justo en la última iteración el modelo alcanzó una precisión del 92% para el conjunto de datos de *training* como los de validación.





Capas convolucionales

Las capas convolucionales son
Los objetos de entrada que se usaron para la visualización fueron tres elementos aleatorios.

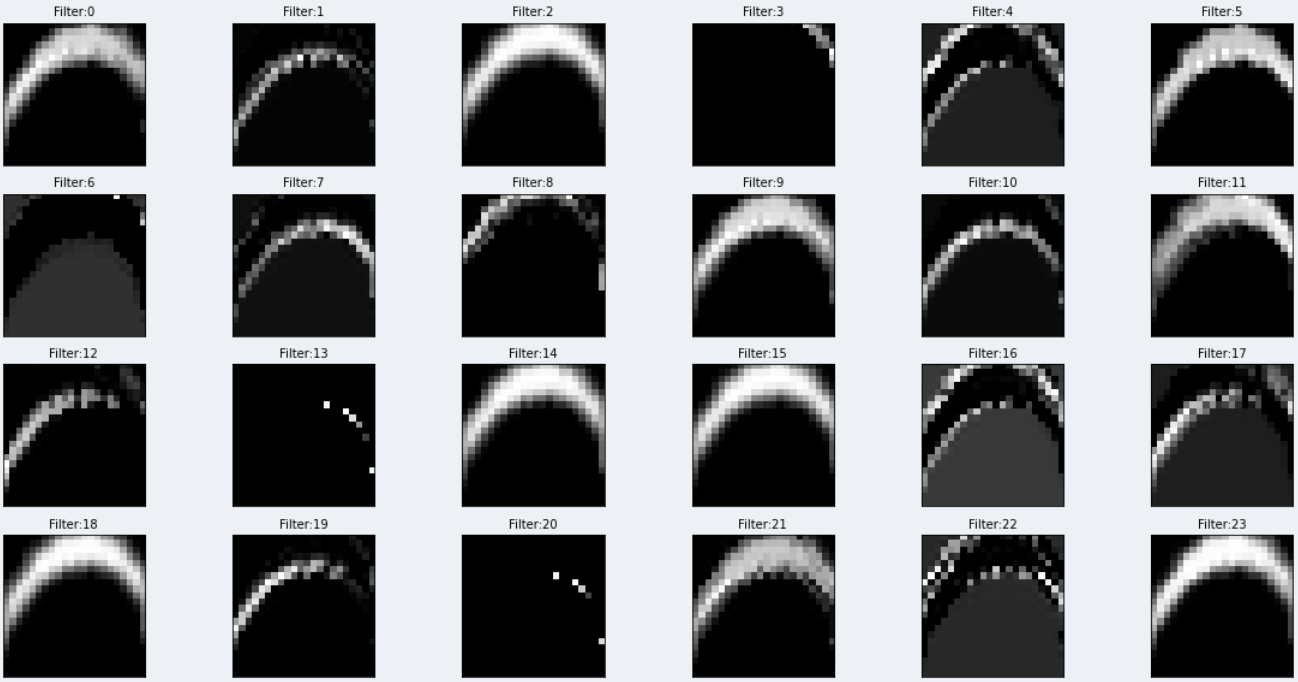


Primera capa convolucional

Los objetos de entrada que se usaron para la primera capa convolucional con un filtro obtuvieron resultados interesantes.

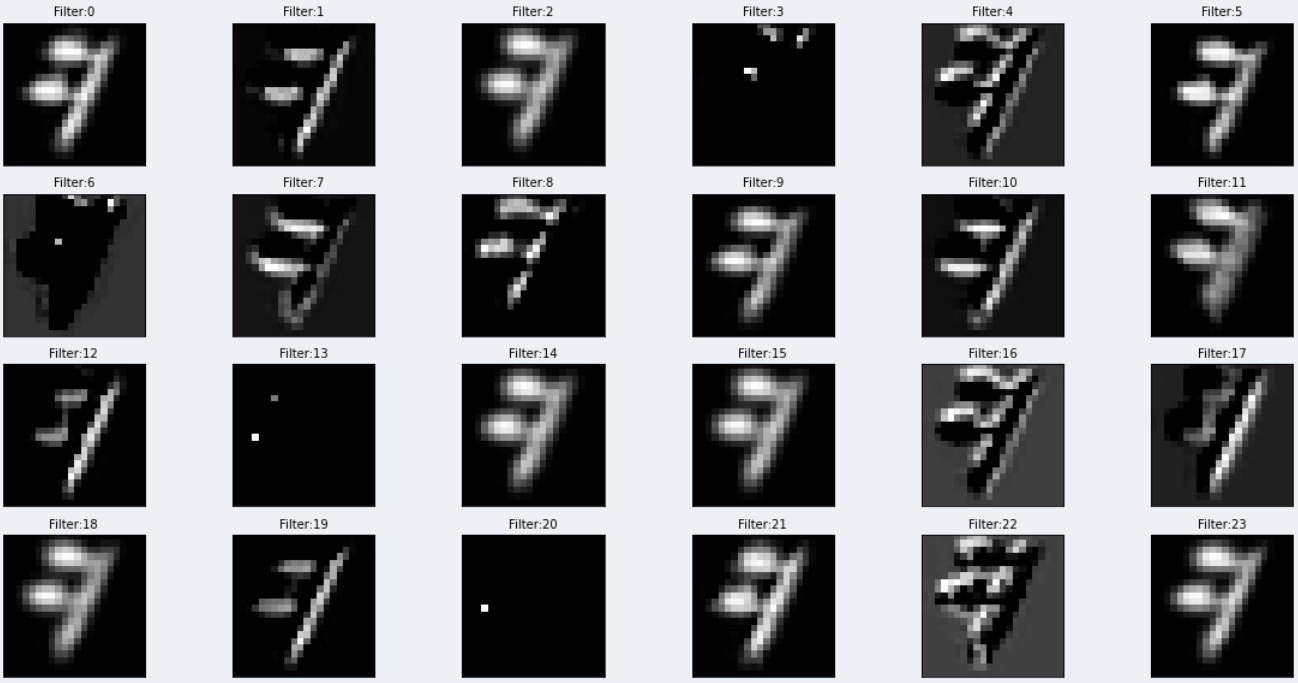
Ligadura

Como se puede apreciar en la primera capa la red neuronal ve la ligadura en su completitud, sobre todo la figura arqueada que la caracteriza. Los filtros 4, 7, 10, 16, 17 y 22 hacen una inversión en los colores del signo quedando lo blanco de color negro. El filtro 3 parece que eliminó la mayor parte de la figura mientras que en los 13 y 20 aún queda un rastro de media curva. En esta capa parece empeñarse en la mitad superior de la imagen.



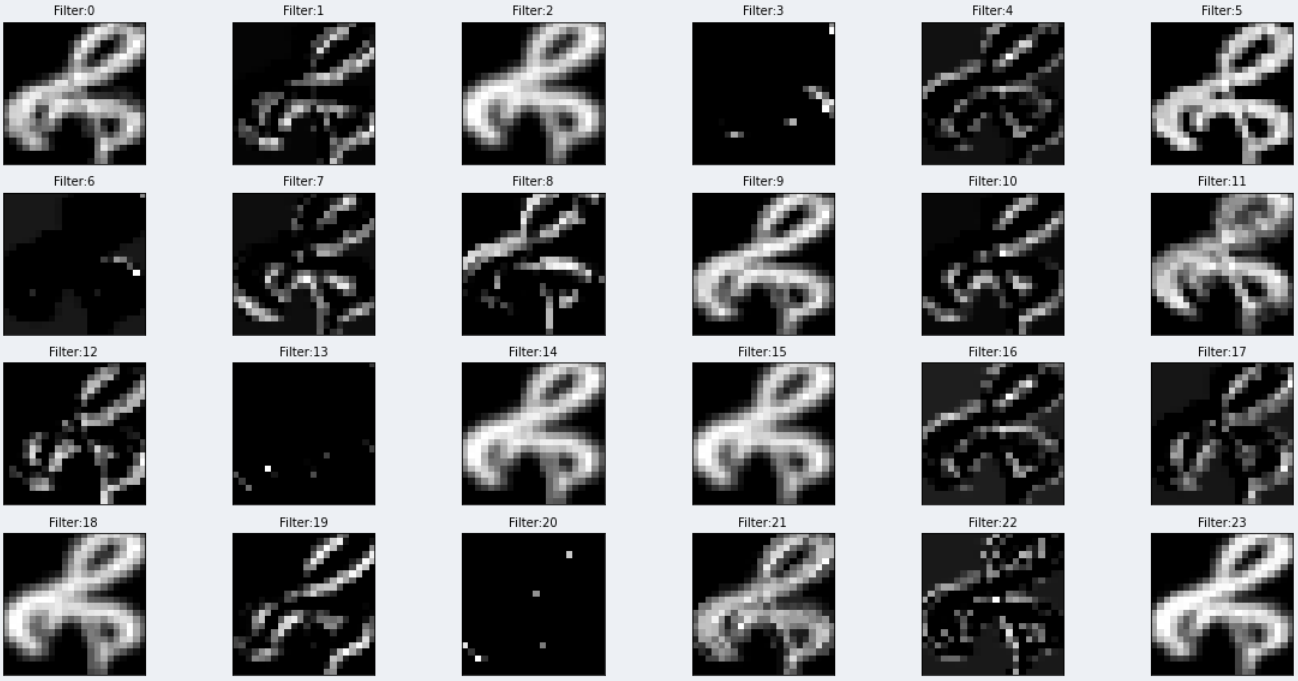
Silencio de semicorchea

En todos los filtros (en su mayoría) puede apreciarse perfectamente la figura del silencio a la perfección. El modelo aparenta tener una excelente forma de reconocer los símbolos en la primera capa convolucional. En los filtros 1 y 19 parece que identifica los trazos base de la plica y las corcheas.



Clave de Sol

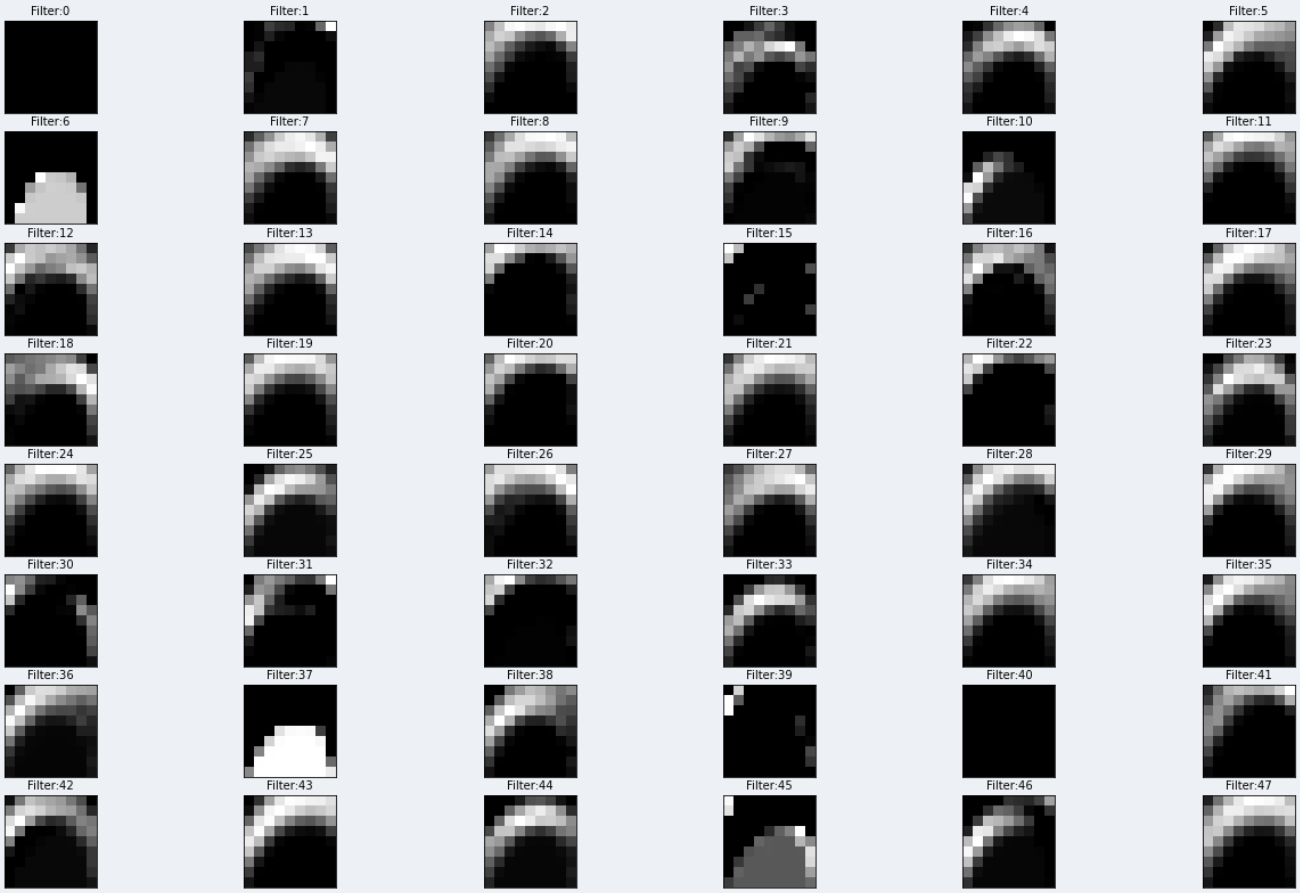
La clave de sol en mi opinión es el elemento más interesante tanto para la primera como para la segunda capa convolucional. La segunda y quinta fila invierten por completo los colores del símbolo, Mientras que los filtros 3, 6, 13 y 20 lo desaparecen. Las demás sólo añaden *blur* o enfocan la imagen.



Segunda capa convolucional

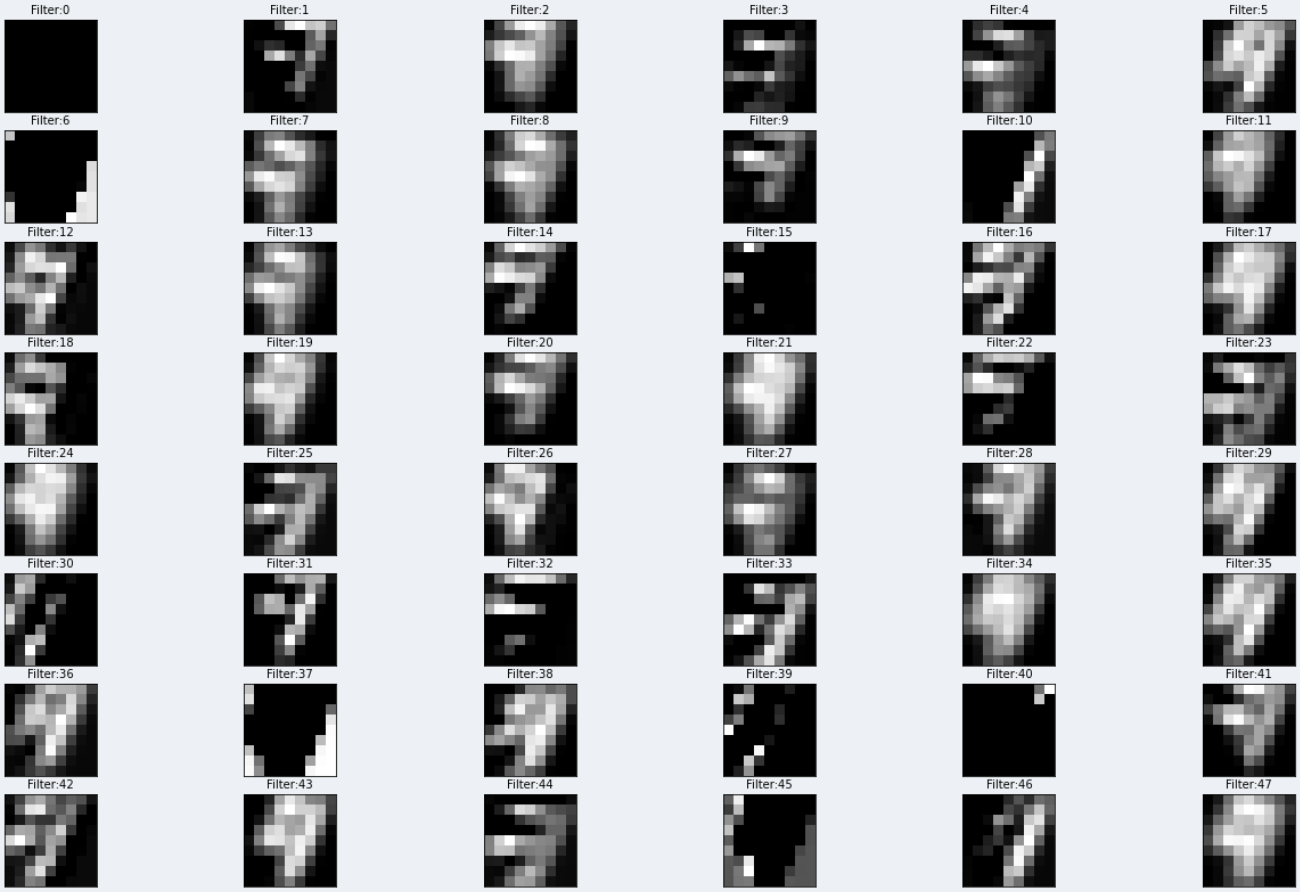
Ligadura

En esta parte de la red neuronal principalmente aprende que hay un gran hueco en la imagen y que la parte de arriba tiene forma curva.



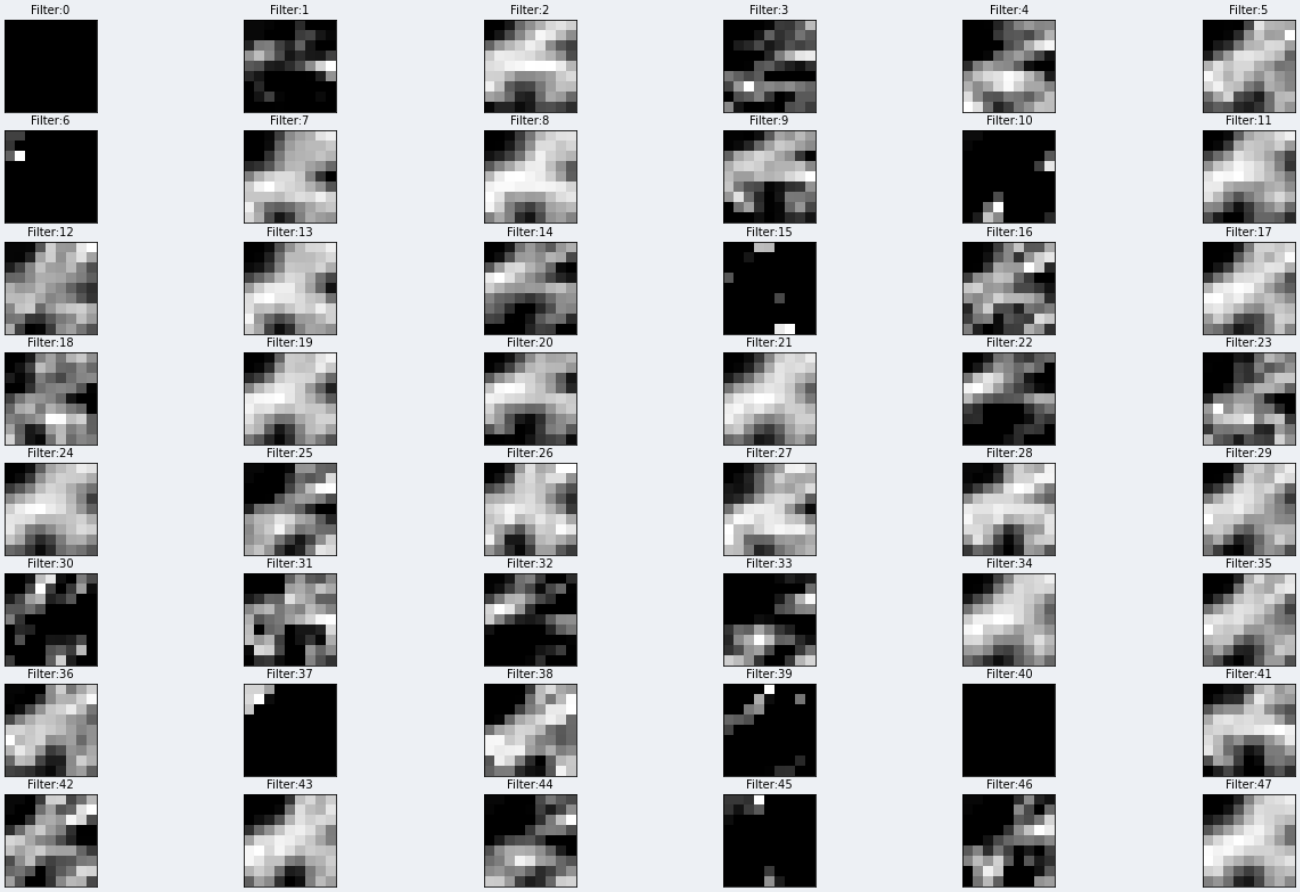
Silencio

El silencio se me asemejó que busca la figura de una F invertida. Las capas 32 y 10 descompusieron al símbolo en trazos básicos. Otra capas añadieron ruido a la imagen.



Clave de sol

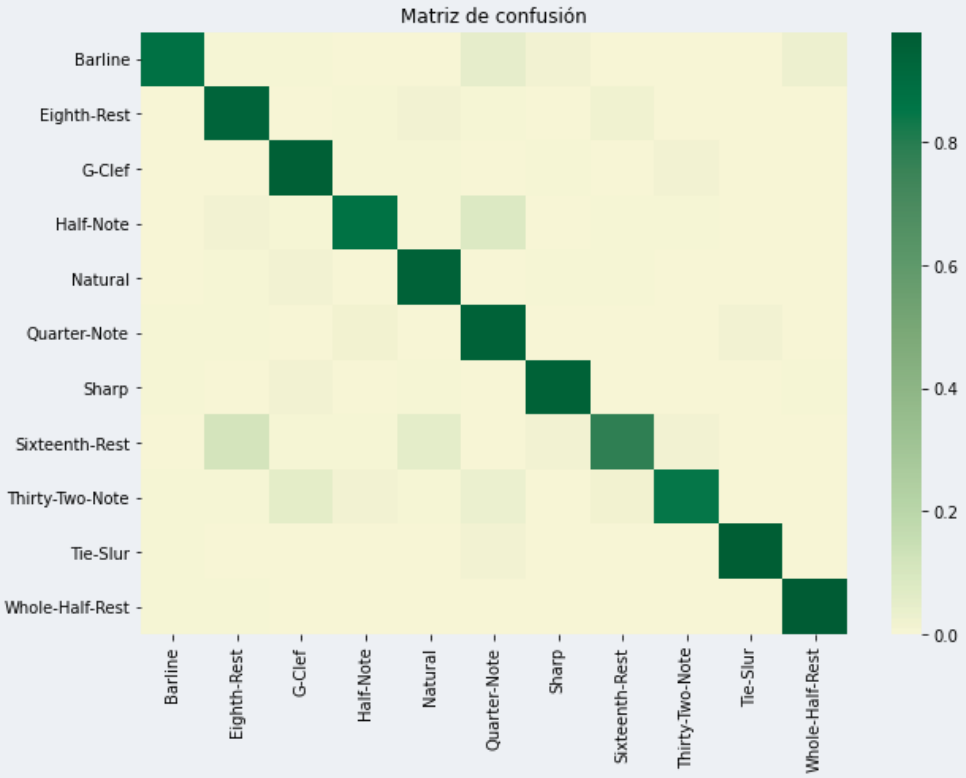
Repito, que de todas esta es la más interesante para mí, pareciera que captara partes de la clave, la cabeza, el cuerpo y trazos finales.



Pareciera que enfoca únicamente el centro de la clave buscando el final achirulado del centro. Los filtros 4, 18, 23 y 44 parecen buscar el pliegue de la cabeza. Los 10, 15 y 37 buscan los espacios vacíos invirtiendo los colores de la imagen.

Resultados del modelo

El modelo se desempeñó bastante bien alcanzando un 93% de precisión y un 28% de pérdida para el conjunto de datos del entrenamiento. En la matriz de confusión para las 11 clases se nota que predijo bien, ya que la diagonal se encuentra marcada y levemente se ven algunos cuadros alrededor con un color raramente alejado.



Modelo con submuestreo

Se hizo un modelo con submuestreo, ya que como se ve en la primera gráfica del informe, las clases están desbalanceadas, habiendo muchos objetos de un tipo y pocos de otros.

Tras entrenar otro modelo con las mismas condiciones y número de capas el modelo se desempeñó mucho peor, en las 20 iteraciones alcanzó un 87% de precisión y un 40% de pérdida en el conjunto de datos de entrenamiento. Por lo que se consideró que la red neuronal aprende mejor con clases desbalanceadas que con unas igualmente distribuidas.

Modelo con todas las clases

Como motivación adicional se entrenó un modelo con 30 clases, y las 39 mil observaciones, el cual obtuvo un buen rendimiento. Se hizo con 60 iteraciones, tres capas convolucionales y dos capas ocultas con activación RELU.

El desempeño del modelo fue con un 95% de precisión con los datos de entrenamiento y 94% de precisión para los datos de testeo. Un 26% de pérdida para el entrenamiento y 16% para el testeo.

CONCLUSIONES Y RECOMENDACIONES

- Probar a entrenar la red con más símbolos resultantes, parece que los datos son bastante buenos para entrenar una diversidad de objetos.
- Usar otras estrategias de DeepLearning para ver su diferencia. De momento este simple modelo arrojó muy buenos resultados.
- El submuestreo no fue efectivo, el imbalance de clases ayuda al modelo a aprender.
- Conseguir datasets con imágenes a color para ver como se sensibiliza el modelo a los colores, en el caso algunas notaciones el color de la nota identifica su duración y expresión.
- Al parecer en la primera capa convolucional el modelo ya aprendió muy bien las figuras de cada uno de los símbolos