

Deep Learning analysis for common pigmented skin lesions classification problem.

Antonio Aguilar
Patricio Barrero
Antonio Hernández
Rodolfo Figueroa
Hugo Moran

Abril 2022

1 Problema

El cáncer de piel es por mucho el tipo de cáncer más común. Casi todos los casos de cáncer de piel pueden ser tratados eficazmente si son detectados en sus etapas iniciales, por este motivo es de suma importancia conseguir modelos que puedan ayudar a la sociedad a detectar estas situaciones con tiempo.

El melanoma es el más mortal de todos los cánceres cutáneos y está ligado fuertemente a mutaciones adquiridas causadas por la exposición a la radiación UV de la luz solar. El melanoma es una neoplasia (masa anormal) relativamente frecuente que puede curarse si se detecta y trata cuando está en sus primeras fases. En Estados Unidos, solo el 5% de los cánceres en la piel es de tipo melanoma, pero es responsable del 75% de las muertes por cáncer cutáneo. Debido a este gran porcentaje de muertes, el cáncer de la piel se divide comúnmente entre melanoma y no-melanoma.

Para los dermatólogos es complicado detectar el cáncer en la piel viendo solo una dermatoscopia, en la mayoría de las veces es necesario realizar una examinación patológica o una biopsia, lo cual es costoso y requiere de tiempo. Así, apoyar al médico con las últimas herramientas computacionales (deep learning) y matemáticas puede ser la diferencia entre detectar un cáncer a tiempo y el no hacerlo.

2 Objetivos

Objetivos generales

Comprender la importancia de prestar atención a las etapas iniciales de las lesiones cutáneas, visualizando el aspecto y la distribución de la misma.

Entender el funcionamiento fisiológico de estas lesiones para estar más preparados a la hora de una examinación propia.

Objetivos particulares

Preprocesar las imágenes para crear un estándar de estas que sirva como base al modelo a usar.

Usar las imágenes anteriores en redes neuronales con el fin de mejorar el accuracy.

3 Metodología

Obtención de datos

Para la obtención de los datos, usaremos el dataset HAM10000, el cual es una colección de 10000 dermatoscopías de diferentes poblaciones. Los casos incluyen una colección representativa de todas las categorías importantes de lesiones cutáneas. Entre estas categorías tenemos: queratosis actínica y carcinoma intraepitelial (akiec), carcinoma basocelular (bcc), keratosis benigna (bkl), dermatofibroma (df), melanoma (mel), nuevos melanocíticos (nv) y lesiones vasculares (vasc).

Preprocesamiento

Una vez obtenido el dataset completo, clasificaremos las imágenes entre dos categorías, los melanomas y las demás, el motivo de esto se explicó al principio del escrito. Una vez hecha esta división, aplicaremos una reducción de canales para quedarnos con las imágenes en grises, normalizaremos los píxeles y estandarizaremos el tamaño de las imágenes.

Modelado

Ya extraídas las características usaremos un modelo de red neuronal bastante usado al tratarse de imágenes, usaremos una red neuronal convolucional, la cual ha probado ser muy útil en la clasificación de imágenes.

Evaluación

En esta parte evaluaremos el modelo, se evaluará la precisión del mismo. Se analizará el desempeño del conjunto de entrenamiento, de validación y de prueba.

Producción

En esta parte se pondrá en producción el modelo. Se creará una pequeña interfaz en donde el usuario pueda subir su imagen para ser clasificada por el modelo.

Todo el flujo de trabajo anterior, lo podemos ver en el siguiente diagrama

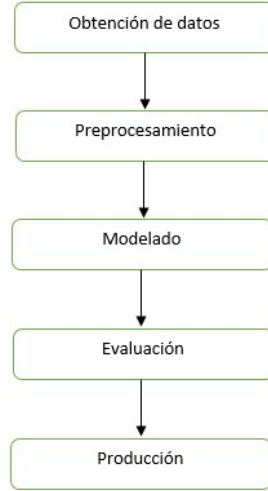


Figure 1: Diagrama del flujo de trabajo del proyecto

4 Estado del arte

Para esta sección se estuvo investigando en varios artículos que se relacionan con el tema, los cuales los dejaremos en la sección de bibliografías. El que más nos llamó la atención es el de *Yao et al*, el cual describiremos un poco con dos imágenes. Primero veremos el flujo de trabajo de este artículo.

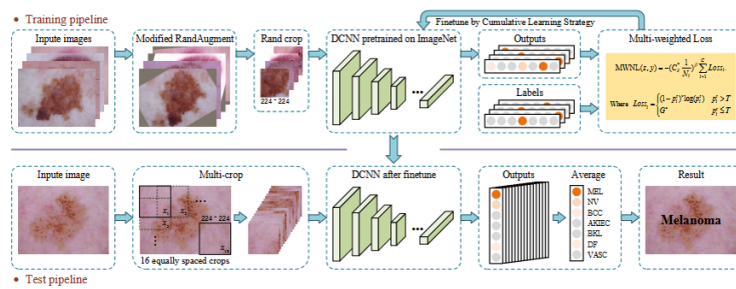


Figure 2: Diagrama del flujo de trabajo del artículo

De forma general este flujo es muy común en ese tipo de problemas, lo que llama la atención es la última parte del pipeline de entrenamiento, el cual describe la función que usaron para ponderar las diferentes clases. El pipeline de evaluación es un poco más familiar para nosotros, en la última parte se utiliza un tipo de softmax para obtener la clase con mayor probabilidad. Ahora veamos un poco la arquitectura de la red usada.

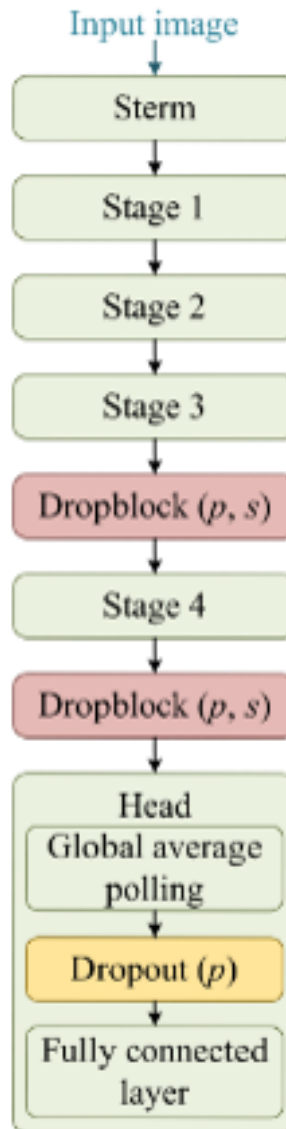


Figure 3: Diagrama de la arquitectura del artículo

Esta arquitectura es como una Regnet, la cual fue bastante complicado de entender, pero hace uso de varias técnicas como la regularización para tener mejor desempeño en el entrenamiento. Nosotros no hicimos cosas tan sofisticadas, pero como veremos más adelante si se obtuvieron buenos resultados.

5 Desarrollo

Preprocesamiento

Después de obtener los datos, se inició el flujo de trabajo con las transformaciones, para dejar las imágenes preparadas para la parte del modelado. Así para esta sección se aplicaron 5 transformaciones: Resize, Random Horizontal Flip, Random Rotation, To Tensor, Normalize. Veamos una breve descripción de estas funciones.

Resize: Con esta función se homologo el tamaño de todas las imágenes.
Random Horizontal Flip: Con una probabilidad dada esta función voltea la imagen en el eje horizontal.
Random Rotation: Con una probabilidad dada esta función rota la imagen x cantidad de grados.
To tensor: Convierte la estructura de los datos en tensores de Pytorch.
Normalize: Ajusta los valores de los píxeles a un rango predeterminado.

Arquitectura

Para la parte de la arquitectura se optó por tomar las CNN, las cuales, como ya se ha visto en las clases tienen muy buenos resultados cuando se trabaja con imágenes. Las características de la red serían:

- 5 capas en total
- 2 convolucionales y 3 lineales
- Función de activación ReLu
- Optimizador Adam
- Función de pérdida cross entropy ponderada

En el siguiente script de código, se muestran algunos parámetros con respecto a cada capa de la arquitectura:

```
self.conv1 = nn.Conv2d(3, 6, (5, 5), padding=2)
self.conv2 = nn.Conv2d(6, 16, (5, 5))
self.fc1 = nn.Linear(16 * 54 * 54, 120)
self.fc2 = nn.Linear(120, 84)
self.fc3 = nn.Linear(84, len(class_weights))
```

Resultados

Después de entrenar el modelo con el GPU otorgado por Kaggle, se guardó el mejor modelo para poder realizar las predicciones correspondientes y sus evaluaciones. Como se ha estado comentando a lo largo del discurso del proyecto, se desarrollaron dos modelos, uno que llamaremos el modelo base (o de referencia) el cual toma en cuenta todas las categorías de las lesiones del dataset usado. Por otro lado, se tiene el modelo binario (melanoma-no melanoma) el cual es un poco más de interés por las causas explicadas al principio del escrito.

Veamos primero la gráfica de la función de pérdida del modelo base

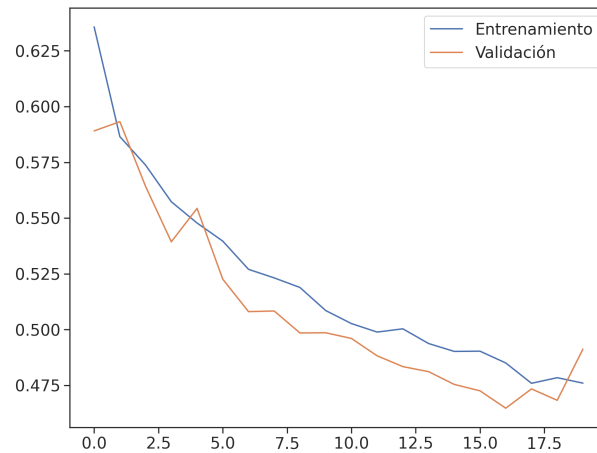


Figure 4: Gráfico de la función de pérdida (modelo base)

Se puede notar que en las épocas mostradas tanto el entrenamiento como la validación parecen converger hasta cierto punto. Todo luce de forma normal como los modelos que ya habíamos hecho en actividades anteriores. Por otro lado veamos el reporte de métricas de este mismo modelo.

	precision	recall	f1-score	support
0	0.23	0.62	0.33	65
1	0.37	0.62	0.47	103
2	0.49	0.23	0.31	220
3	0.20	0.17	0.19	23
4	0.27	0.59	0.37	223
5	0.91	0.66	0.76	1341
6	0.31	0.79	0.45	28
accuracy			0.60	2003
macro avg	0.40	0.52	0.41	2003
weighted avg	0.72	0.60	0.63	2003

Figure 5: Reporte de métricas (modelo base)

Las clases serían las siguientes:

- (0) "Queratosis actínica"
- (1) "Carcinoma de células basales"
- (2) "Lesiones benignas similares a queratosis"
- (3) "Dermatofibroma"
- (4) "Melanoma"
- (5) "Nevo melanocítico"
- (6) "Lesiones vasculares"

La clase que mejor clasifica el modelo es la clase 5 (Nevo melanocítico) lo cual hace sentido ya que es la clase con mayor representación del dataset. Si bien los resultados no son malos, falta bastante trabajo por hacer para obtener un modelo óptimo.

Por otro lado, veamos ahora los resultados para el modelo binario, igual que en el modelo anterior, mostraremos primero la gráfica de la función de pérdida.

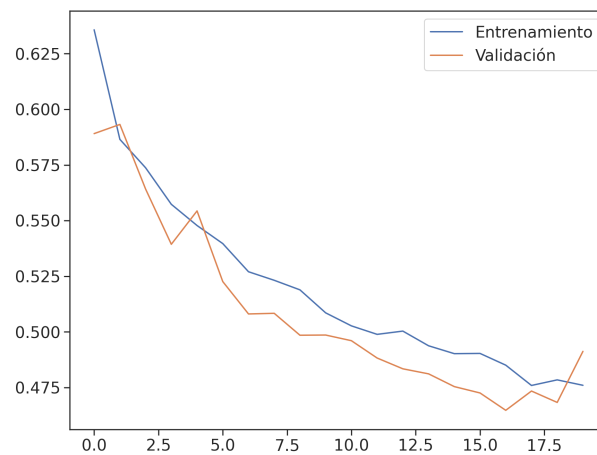


Figure 6: Gráfico de la función de pérdida (modelo binario)

Se puede observar un comportamiento similar al modelo base, veamos ahora la parte más importante de esta sección, que es el reporte de métricas del modelo binario

	precision	recall	f1-score	support
0	0.95	0.74	0.83	1780
1	0.25	0.71	0.37	223
accuracy			0.73	2003
macro avg	0.60	0.72	0.60	2003
weighted avg	0.87	0.73	0.78	2003

Figure 7: Reporte de métricas (modelo binario)

En este tipo de modelos, la métrica que más nos interesa es la de "recall" ya que es la que nos dice que tanto porcentaje de las personas que sí padecen de melanoma las está detectando el modelo, en estas situaciones es preferido que se mal clasifique a personas que no tienen melanoma como si lo tuvieran a que lo haga con las personas que sí tienen melanoma y parezca que no lo tienen.

Así, podemos ver que nuestros modelos tienen un desempeño bastante aceptable, ya que la clase de melanoma tiene un recall de 0.71, es decir, no está detectando el 71% de las personas que padecen de melanoma.

Deployment

Una vez guardado el checkpoint del modelo, se pasó a realizar la producción del mismo, se usó la herramienta de streamlit para facilitar la creación de la app, el link para la aplicación sería el siguiente:

<https://share.streamlit.io/rodolfofigueroa/skinlesionsapp/main>

La interfaz sería la siguiente

Sube un checkpoint

Drag and drop file here
Limit 200MB per file • CKPT

Browse files

Sube una imagen

Drag and drop file here
Limit 200MB per file • JPG

Browse files

☐ Binario ☐ Evaluación

Evaluar

Figure 8: Interfaz de la aplicación

Se es necesario subir dos archivos, el primero de ellos es el modelo cargado y el segundo es la imagen a predecir. Posteriormente se tienen dos opciones, una que le dice a la aplicación si la predicción será binaria o será de todas las clases y por último una opción que permite visualizar la etiqueta de la imagen subida.

6 Conclusiones

Como se pudo apreciar en la presentación del deployment, algunas imágenes no eran clasificadas correctamente, y de hecho las categorías con mayor precisión fueron muy diferentes a la clase real de la imagen. Una de las problemáticas que detectamos fue que en algunas clases el número de representantes era muy bajo, y por otro lado en unas dos clases el número de elementos eran mucho mayor que las demás, generando así un desbalanceo en las clases.

Aún con las complicaciones de los datos, creemos que hicimos un buen papel a la hora de preparar las imágenes y de elaborar la arquitectura para ambos modelos. Los resultados obtenidos son complacientes, si bien nos gustaría tener un poco más de imágenes en algunas clases, estamos satisfechos con el desarrollo del proyecto.

7 Bibliografía

- Yao, P., Shen, S., Xu, M., Liu, P., Zhang, F., Xing, J., Shao, P., Kaffenberger, B., Xu, R. X. (2022). *Single Model Deep Learning on Imbalanced Small Datasets for Skin Lesion Classification*. In *IEEE Transactions on Medical Imaging* (Vol. 41, Issue 5, pp. 1242–1254). Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/tmi.2021.3136682>
- Masood, A., Ali Al-Jumaily, A. (2013). *Computer Aided Diagnostic Support System for Skin Cancer: A Review of Techniques and Algorithms*. In *International Journal of Biomedical Imaging* (Vol. 2013, pp. 1–22). Hindawi Limited. <https://doi.org/10.1155/2013/323268>
- Kadampur, M. A., Al Riyaei, S. (2020). *Skin cancer detection: Applying a deep learning based model driven architecture in the cloud for classifying dermal cell images*. In *Informatics in Medicine Unlocked* (Vol. 18, p. 100282). Elsevier BV. <https://doi.org/10.1016/j.imu.2019.100282>
- Ali, M. S., Miah, M. S., Haque, J., Rahman, M. M., Islam, M. K. (2021). *An enhanced technique of skin cancer classification using deep convolutional neural network with transfer learning models*. In *Machine Learning with Applications* (Vol. 5, p. 100036). Elsevier BV. <https://doi.org/10.1016/j.mlwa.2021.100036>
- BINDER, M., STEINER, A., SCHWARZ, M., KNOLLMAYER, S., WOLFF, K., PE-HAMBERGER, H. (1994). *Application of an artificial neural network in epiluminescence microscopy pattern analysis of pigmented skin lesions: a pilot study*. In *British Journal of Dermatology* (Vol. 130, Issue 4, pp. 460–465). Wiley. <https://doi.org/10.1111/j.1365-2133.1994.tb03378.x>
- Tschandl, P., Rosendahl, C., Kittler, H. (2018). *The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions*. In *Scientific Data* (Vol. 5, Issue 1). Springer Science and Business Media LLC. <https://doi.org/10.1038/sdata.2018.161>