

Se quiere hacer una red siameses junto a un algoritmo de logica difusa para poder asegurar que las imagenes fueron clasificadas correctamente. Al final se dejan los papers vistos (no todos se leyeron del todo ya que no es 100% necesario es solo para una idea).

A continuacion les dejo un **TODO** para que vayan viendo que es lo que falta, ademas les dejo otras notitas de cosas que he hablado con Jorge de TE y algunas cosas que podrian servirles para los que nunca han usado ciertas cosas.

## 1 TODO

- Relizar presentacion para antes del 26 de marzo, la cual consiste en presentar los avances de lo que se lleva actualmente del proyecto.
- Tambien, falta pulir los detalles para poder tener un mejor modelo. Hay que ajustar los parametros para que el modelo sea mucho mejor al momento de testear las imagenes.
- Una vez obtenido un buen modelo, lo siguiente es implementar la logica difusa. La cual es un poco complicada en concepto pero sencillamente de implementar gracias a la librerias que existen. Investiguen que librerias existen y elijan la que mejor les parezca, deben de tener en cuenta varias cosas compatibilidad y escalabilidad. Vean si la libreria es activa en cuestion de actualizaciones para asi evitar problemas a futuro. Los pasos que debemos seguir son:
  1. ¿Qué es la Logica Difusa? ¿Para qué la vamos a utilizar? ¿Hay algun beneficio sobre no usarla?
  2. Implementación de código, tendrán que ver como se implementa y utilizar buenas prácticas, utilicen todas las herramientas que necesiten para eso están, si ocupan usar Chat GPT o Deepseek úsenlos pero primero comprendan como funciona el código.
  3. Documentación, debemos documentar en prosa y en la libreta el código escrito por ustedes
  4. Pruebas finales y observaciones.

## 2 Reuniones

- Reuniones todos los miercoles a las 11:00 AM.

### 2.1 Notas:

1. **Semana Marzo 10-14**
2. Actualmente se tiene un prototipo de como esta funcionando una red siameses para poder hacer la clasificacion. El único problema que surgió era el tema del dataset ya que como es pequeño podría resultar en resultados malos, ahí fue donde propuse el que nosotros realizar la clasificaciones pero hablando con Waissman me dijo que existe una forma que en base a un pequeño dataset poder realizar una mejor clasificacion y es usando basicamente dos imagenes como referencia (esas imagenes son del dataset), lo cual en vez de hacer comparacion 1 a 1, seran de 1 a 2 lo cual permite crecer mucho mas nuestro dataset a eso se llama Triplet Loss Network, y será mas grande al hacer el data augmentation. Se hizo una implementacion muy basica la cual falta poder ajustar los parametros para obtener mejores resultados.

3. Se hizo en libreta de jupyter para asi poder tenerla en github, pueden subir la libreta a collab si es que no tienen una pc que pueda soportar correr el entrenamiento (no es muy pesado, pero si no es una pc un poco potente puede tardar mas). Ya que jupyter lab corre en local usando recursos propios.
4. Se actualizó la libreta ya que existian unos problemas al momento de entrenar. Lo cual ocasionaba que la perdida del entrenamiento fuera muy grande y con ello afectara el resultado.
5. Se hizo una implementacion de una funcion la cual descarga de un servidor el dataset para asi no tenerlo en el repositorio todas las imagenes y ocasione que sea tardado hacer push y pull ademas de evitar errores por multiples modificaciones.

Algunos comandos que les puede servir y herramientas:

- Editores de codigo:
  - Visual Studio Code
  - PyCharm
- Herramientas necesarias:
  - Python 3.13.2
  - Git
- Comandos Git:
  - `git brach -m nombre` (crea una nueva rama)
  - `git checkout nombre` (cambias de rama)
  - `git add archivo1 archivo2 carpeta` (se puede agregar lo que sea o tambien usar el `.` para especificar que todo se va a agregar)
  - `git commit -m "Texto"` (aqui se pone todos los cambios que hicieron)
  - `git push` (este es para subir todos los cambios que hiciste)
  - `git pull origin main` (se debe hacer siempre que vayas a hacer cambios y que no tengas modificaciones)

Papers:

- [https://github.com/vbelz/Outliers\\_detection](https://github.com/vbelz/Outliers_detection)
- <https://dl.acm.org/doi/10.1145/3688394#sec-3>
- <https://www.sciencedirect.com/science/article/pii/S0925753521003714>
- [https://www-pub.iaea.org/MTCD/Publications/PDF/te\\_538\\_web.pdf](https://www-pub.iaea.org/MTCD/Publications/PDF/te_538_web.pdf)
- <https://www.ibm.com/think/topics/shedding-light-on-ai-bias-with-real-world-examples>
- <https://dataheroes.ai/blog/outlier-detection-methods-every-data-enthusiast-must-know>
- <https://github.com/13muskanp/Siamese-Network-with-Triplet-Loss>
- <https://github.com/Ekeany/Siamese-Network-with-Triplet-Loss>
- <https://ieeexplore.ieee.org/document/9353191>