

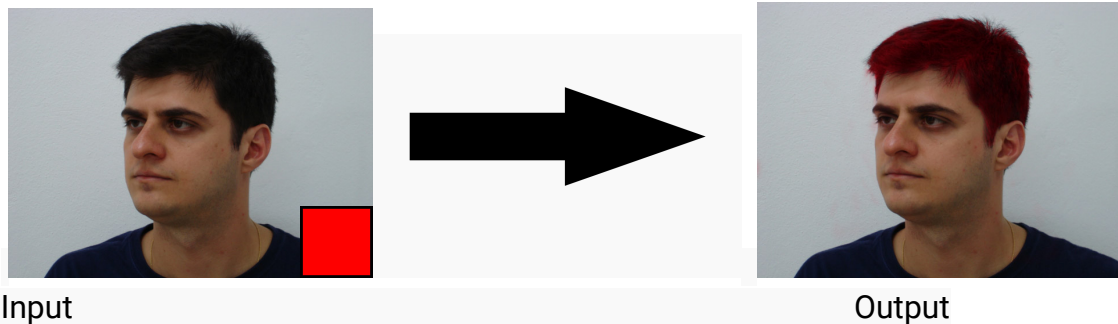
Introducción:

Este proyecto ha sido creado para el concurso de dot csv #RetoDotCSV2080Super (0) esto indica que se encuentra abajo en referencias, contendrá un numero dentro de los corchetes para poder identificarse.

Objetivo del proyecto:

Según las bases del concurso había que crear un proyecto que cogiera una imagen de input y que sacase otra imagen de output, entonces el problema planteado en este trabajo es el siguiente:

Las imágenes de input serán imágenes de personas con un cuadrado de un color, y ese cuadrado con una linea negra en los bordes, los bordes son de 4 pixels y el cuadrado de 128 pixels por lado(hay que tener en cuenta que este tamaño en pixels es en las imágenes inicial, después cuando son cargadas en el resize cambia) . Las imágenes de output serán las mismas imágenes de las personas de las imágenes de input pero con el pelo del color del cuadrado de las imágenes de input.



Entonces, lo que haría nuestro modelo es cambiar el pelo a las personas con el objetivo de ver como le queda el pelo de otro color.

Aparte de aprender a cambiar el color de pelo, el programa intentará predecir que va donde está el cuadrado de la imagen de input, aparte de aprender a identificar el pelo. Un objetivo secundario sería que en vez de poner una color en el cuadrado del input poner un degradado de color o una textura para ver si puede trabajar igual que si fuera un color.

Programar: El dilema que he tenido a la hora de programar es el siguiente: la idea principal era trabajar con pix2pix HD(0) que gracias ha su modelo escalable puede trabajar con imágenes más grandes y además con más calidad, todo esto gran parte debido a las redes residuales(rk) que implementa, el problema ha estado en que yo

nunca he programado en python y menos en las librerías de tensorflow y keras,, entonces decidí empezar haciendo el modelo de pix2pix siguiendo el tutorial de dot csv (1), ese proyecto me funciono y al final decidí que sería el modelo que implementaría, esto debido a que mis intentos de crear un modelo pix2pix HD fracasaron, hubo un proyecto de github que con él casi puedo implementar pero al estar en tensorflow v1 no lo llegué a comprender del todo y por no saber como implementarlo decidí abandonarlo. (el proyecto de pix2pix HD mencionado : 2).

Este modelo comparte la estructura de los módulos para construir el modelo, ejemplo: uk, rk, c7s1-k, dk, ck. Pero a la hora de formar el modelo utiliza una mezcla de los módulos diferente a la de pix2pix HD.

Dataset:

El dataset ha sido cogido principalmente de (3), también he usado esta página web para buscar los datasets (4).

A la hora de editar el dataset se han usado las acciones de photoshop, para las imágenes del input usando las especificaciones anteriormente (4 pixels los bordes negros, y el cuadrado de 128 pixels), esto gracias a la herramienta de formas de photoshop.

Para las imágenes de target se cogía la primera de la lista, se le añadió las siguientes capas ==> niveles(en esta se crea una mascara, y las siguientes capas quedarían enlazadas a esta), curvas tono/saturación, la configuración dependerá de gustos, la saturación la dejé ha cero y los otros valores no lo modifique demasiado, la última capa añadí fue la de color uniforme.

Entonces el siguiente paso era que la mascara de la capa de niveles tuviera en blanco pintado el pelo y lo otro en negro, cuando acababa de hacer eso con una imagen, guardaba esta y después copiaba la siguiente imagen y la pegaba en la donde estaba la primera imagen para editarla desde ahí, esto no es lo más recomendable si no se hace de la forma adecuada, este paso yo lo hizo de la forma incorrecto porque al cambiar de imagen no borraba de forma exhaustiva la mascara para el pelo de la anterior imagen y por eso en el dataset hay espacios en las imágenes con un color que no deberían,la forma correcta habría sido o aplicar las acciones de fotoshop para que pusieran las capas anteriormente mencionadas en todas las imágenes o otra opción hubiera sido borrar y poner otro o resetear la mascara(si se puede).

Después de este paso se realizó acciones en todas las imágenes del target, el numero de acciones realizadas dependía de cuantos colores quería porque las acciones realizadas consistían en cambiar el color de la capa de color uniforme y guardar la imagen.

La carpeta Input1 y HF_Color1 porque los nombres no son iguales.

Se puede acceder al dataset por mi drive(6).

Código:

El código de este proyecto es muy parecido del video(2) porque ha sido copiado de ahí sufriendo algunas pero muy pequeñas modificaciones, lo puedes encontrar como pix2pix en el repositorio, y aquí te comparto el enlace en google colab = (5).

Pero este código hay que decir que esta configurado con mis carpetas de google drive.

Entrenamiento:

El modelo ha sido entrenado en google collab, esto a limitado todo el proyecto porque cada una rato por cualquier motivo el proyecto se desconectaba de los entornos alojados de google dificultando la tarea, por este motivo se intento usar servidores locales, aunque hubiera sido con menos potencia había esperanza que fuera más estable trabajar con eso, pero igualmente no lo acabamos de conseguir.

El proyecto ha sido entrenado con varias carpetas una para cada color, por ese mismo motivo se uso está linea de código para coger siempre las mismas imágenes:

`"np.random.seed(23)".`

Pero usando este método no acabo de funcionar bien y entonces puse todos los inputs en una carpeta y todos los targets en otra carpeta, el inconveniente de esto es que las imágenes que muestre el generador de imágenes lo más probable es que estén siendo entrenadas pero en otro color.

En algunas etapas del entrenamiento el proyecto actuó como no debía, dando errores donde no los había o al menos yo no los veía, por eso en el proyecto hay alguna linea de código repetida, porque estás lineas automáticamente me solucionaron el problema (imagen1).

El proyecto ha sido entrenado distintas veces por culpa del problema de las carpetas, y por cuestión de tiempo, no ha podido ser entrenado debidamente en el último entrenamiento, seguramente porque en el momento que escribo esto lo estoy "acabando de entrenar" el proyecto no va a poder ser entrenado mucho más de 100 épocas, se recomienda que se vuelva ha entrenar para poder ver mejores resultados, también hay que especificar que también por el tiempo solo ha entrenado con cuatro colores distintos, como tengo más colores hechos los dejaré disponibles en el repositorio por si alguien quiere mejor el proyecto.

Test:

En el código después de `generate_images` pone `#` Para hacer el testing ejecuta este código , lo tienes que ejecutar después de haber ejecutado todas las otras celdas incluidas la de `train` (la de abajo del todo), cuando ejecutas la de `train` esperas a que se cargue, y cuando este ejecutándose en epoch 0, interrumpes la celda y ejecutas la celda que pone `#` Para hacer el testing ejecuta este código, recuerda que esta se encuentra después de `generate_images`.

Conclusiones:

Por desconocimiento de google colab no sé si el modelo fue entrenado de la forma más correcta, los resultados no son los más deseados y se podrían mejorar con el modelo pix2pix HD, pero son unos resultados que muestran .

¿ Por qué es interesante mi proyecto?

Este proyecto en vista general es interesante porque experimenta con el machine learnig para hacer que esta pueda modificar imágenes gracias ha inputs que se encontrarían en el mismo input, pero la idea que tuve para empezar este proyecto considero que es lo más interesante de este, que sería enseñarle ha la inteligencia artificial que es un color, la idea inicial era añadiendo una etiqueta en el input pero por no saber hacerlo lo hice con esta medida ingeniosa.

Referèncias:

0, paper pix2pix HD: <https://arxiv.org/pdf/1711.11585.pdf>

1: <https://www.youtube.com/watch?v=YsrMGcgfETY&t=2457s>

2: <https://github.com/yikaiw/CycleGAN>

3: <https://fei.edu.br/~cet/facedatabase.html>

4: <http://www.face-rec.org/databases/>

5: https://drive.google.com/file/d/1H8-NtQpc2MkRsdTqP8nBGGvJpy_PipKB/view?usp=sharing.

6: <https://drive.google.com/drive/folders/1eZXYVgqKNoqBlybLn6GVISL74Tlj7EVx?usp=sharing>