

Procesamiento de imágenes utilizando Deep learning y Carla Simulator

Gatica, Isaís

email: isaiasgaticai@gmail.com

Saez, Lautaro Andres

email: lautaroandressaez@gmail.com

Resumen—En este informe se presenta la implementación de una red Pix2Pix capaz de pasar de una imagen RGB a una imagen segmentada donde cada color indica un objeto diferente. Se comenzara desde cero obteniendo el dataset mediante el simulador CARLA para luego poder entrenar la red con una gran variedad de imágenes con diferente clima y vegetación. Finalmente se probara con imágenes reales para analizar la viabilidad de entrenar una red con un simulador y luego ingresarla en el mundo real.

I. INTRODUCCIÓN

En los últimos años, debido a la necesidad de resolver cada vez más complejos y la posibilidad de utilizar GPU's las redes neuronales han cobrado gran importancia en nuestras vidas, y ya son parte del día a día, cosas tan simples como los filtros de Instagram basan su reconocimiento facial en redes neuronales.

Las redes permiten resolver problemas que van desde la clasificación de imágenes hasta la conducción autónoma, claro está que para esto existen diferentes arquitecturas las cuales son específicas para determinadas tareas, como el procesamiento de imágenes (CNN, GAN, YOLO), procesamiento de texto (RNN), procesamiento de datos (NN).

La desventaja principal de estos modelos supervisados es la necesidad de un dataset muy bien estructurado y para el cual es necesario el trabajo humano, esta es la tarea más importante y que si está mal realizada el red no tendrá un desempeño correcto.

II. MARCO TEÓRICO

II-A. Deep Learning

Las redes neuronales son muy utilizadas en el campo de la minería de datos, debido a su gran versatilidad. Dentro de este mundo existen 2 grandes tipos de entrenamientos para los supervisados que dada una entrada X se conoce la salida deseada Y y otros casos los algoritmos no supervisados como los modelos de Q-learning o los mapas autoorganizados (SOM). En este proyecto se optó por un entrenamiento supervisado.

Para el modelado de la red neuronal se utilizó un modelo llamado Pix2Pix [4]. El cual permite dada una imagen de entrada X obtener una imagen de salida Y haciendo uso de una arquitectura U-Net [3].

Este modelo cuenta con 2 redes neuronales una denominada el generador en la cual recibe la imagen X y debe ser capaz de aprender a transformarla en la imagen \hat{Y} , la cual es una aproximación de la imagen Y . Y por otro lado el discriminador el cual debe tomar la imagen \hat{Y} para lograr reconocer las partes

de esta que sean erróneas. La salida del discriminador debe ser una imagen resucida en tamaño que cuente con un único canal de color, siendo esta imagen una cuantificación del error de la red generativa.

Este problema desemboca en una pelea entre el generador ya que debe aprender a engañar al discriminador y el discriminador que debe lograr siempre reconocer que la imagen \hat{Y} es totalmente falsa, ya que es una creación de la red generativa.

II-B. Preprocesado de los datos

Un paso fundamental para trabajar con algoritmos de imágenes es pasar del dominio $[0; 255]$ a un dominio más acotado, para este trabajo se utilizó el dominio de $[-1; 1]$. De esta forma se logra una mayor eficiencia computacional.

Debido a la gran variedad de dimensiones de las imágenes es recomendable hacer un *resize* a dimensiones más pequeñas para disminuir el uso de RAM cuando son cargadas en memoria y el uso de GPU al realizar las convoluciones. En este caso las imágenes de entrada y de salida son redimensionadas a 256×256 .

III. DESARROLLO

III-A. Obtención del dataset

Para obtener el dataset se colocaron una cámara RGB y una cámara de segmentación semántica [2], ubicadas en la misma posición. Se recolectaron los datos cada 5 segundos. Las condiciones climáticas utilizadas fueron aleatorias, lo que permite obtener un dataset mucho más variado.

Para lograr un manejo autónomo del vehículo se utilizó el autopilot [1]. Con la finalidad de aumentar la variedad de imágenes se utilizaron todos los mapas posibles y se adicionaron otros vehículos autónomos.

III-B. Entrenamiento de la red

Como fue mencionado en el marco teórico el modelo implementado se denomina Pix2Pix [4]. Se utilizó como entrada la imagen de la cámara RGB y la salida será una imagen segmentada, para ello se utilizó el dataset que contaba con casi 8000 imágenes.

III-B1. Tratamiento de las imágenes: Para evitar problemas en el entrenamiento de la red es muy usual llevar los valores de cada canal al intervalo $[-1; 1]$, esto evita problemas de continuidad en red.

Otro paso previo es agrandar la imagen a un tamaño mayor y luego realizar un recorte aleatorio, esto es con la finalidad de realizar un incremento virtual del dataset. Este metodo permite evitar el overfitting. Adicionalmente es posible realizar un flip de la imagen aleatoriamente.

El dataset fue dividido en 2 partes una utilizada para el entrenamiento, aproximadamente el 80 % de las imagenes y otra parte para el testing de la red utilizando el 20 % restante.

Luego de **400** iteraciones se presentan los resultados en la fig.1

Figura 1. Imagenes obtenidas luego del entrenamiento

IV. CONCLUSIONES

REFERENCIAS

- [1] "Carla api documentation," https://carla.readthedocs.io/en/latest/python_api/.
- [2] "Carla sensors reference," https://carla.readthedocs.io/en/latest/ref_sensors/.
- [3] "U-net architecture," <https://en.wikipedia.org/wiki/U-Net>.
- [4] Phillip.I, J.Zhu, T.Zhou, and A.A.Efros, "Image-to-image with conditional adversarial networks," 2018.