Abstract

Se construyo dos redes neuronales convolucionales binarias, una para caras de personas y otra para sombreros, con el fin de localizar caras y sombreros de forma individual en una imagen en particular. Se proceso los resultados de las dos redes neuronales y con cada respuesta se construyo un mapa lógico en el cual se puede localizar la zona donde esta presente el objeto de interés, caras o sombreros. Posteriormente se procesan los dos mapas y del procesamiento se puede etiquetar la imagen indicando si las caras presentes en la imagen están usando o no sombreros.

1 Introducción

Este documento se organizo de la siguiente forma: En la sección 2 se analizan trabajos previos de reconocimiento del contenido de una imagen. En la sección 3 se describen los materiales y metodologías usados para abordar el proyecto. En la sección 4 se profundiza en la metodología usada y en la sección 5 se presentan los resultados obtenidos de esta metodología. En la sección 6 se presentan las conclusiones de los resultados obtenidos.

2 Trabajo Previo

Google Cloud Vision API Image Recognition Tensorflow Integrating Humans and Computers for Image and Video Understanding How Will Google "Read" & Rank Your Images in the Near Future

- Google Cloud Vision API
- Image Recognition Tensorflow
- Integrating Humans and Computers for Image and Video Understanding
- How Will Google "Read" & Rank Your Images in the Near Future

3 Materiales y Métodos

3.1 Materiales

Las imágenes de las caras y otras imágenes diferentes a caras fueron obtenidas de "Labeled Faces in the Wild Home y Caltech 101". Las imágenes de los sombreros y otras imágenes diferentes a sombreros fueron obtenidas de "ImageNet" Se uso la biblioteca TensorFlow para el lenguaje de programación Python 3.5

3.2 Métodos

Se uso una metodología basada en redes neuronales convolucionales tomando como base la implementación de Cifar-10, el cual se encarga de clasificar y etiquetar imágenes RGB de 32x32 de 10 clases diferentes. Debido a que no es viable entrenar una red neuronal con imágenes sin ningún tipo de procesamiento por la gran cantidad de píxeles que implica una imagen, es necesario reducir esta cantidad sin perder, en gran medida, la información importante que brindan todo el conjunto de datos, por lo tanto aplicamos la convolución para resaltar y extraer la información importante de la imagen y eliminar píxeles que no brinden información de interesante, sin embargo la matriz resultante aun es grande para la red neuronal. Para reducir las dimensiones de esta matriz se recurre a la operación no lineal max-pooling que consiste en tomar los valores mas significativos en múltiples ventanas de determinada dimensión de la matriz procesada y como resultado obtenemos una matriz mucho mas reducida y adecuada para el procesamiento en la red neuronal. Es decir, una red neuronal convolucional consiste en dos procesos, el primero se encarga de reducir las dimensiones de los datos a partir de convoluciones y otras operaciones no lineales y el segundo consiste en la red neuronal alimentada con los datos del primer proceso.

La arquitectura de Cifar-10 consiste en una capa de convolución seguida de una operación max-pooling y una normalización, luego otra capa de convolución, una normalización y un max-pooling y finalmente el procesamiento de la red neuronal.

4 Experimentos

4.1 Entrenamiento

Procesar el dataset: se procesa las imagenes de caras y sombreros para formar dos archivos binarios por lotes(aka databatch), cada uno con 10000 imágenes a color y con dos clases diferentes por archivo. La construcción del databatch se basa en el proceso descrito en Cifar-10, es decir, se construye una matriz donde cada una de sus filas corresponde a un vector de 3073 elementos que representan la información presente en la imagen. Para conformar este vector de 3073 elementos primero se debe redimensionar la imagen original a 32x32 conservando las proporciones de la imagen original, luego el primer elemento del vector corresponde a la etiqueta o clase de la imagen procesada (generalmente cero o uno), los siguientes 1024 elementos corresponden a la intensidad de los píxeles del canal r, los siguientes 1024 al canal g y finalmente los últimos 1024 al canal b del espacio de color rgb. Por lo tanto se forma la siguiente estructura de n-imágenes x 3073:

<1 label imagen 1><1024 r><1024 g><1024 b>
<1 label imagen 2><1024 r><1024 g><1024 b>
. . .
<1 label imagen n><1024 r><1024 g><1024 b>

4.2 Pruebas

Para evaluar una imagen, es necesario realizar un procedimiento que, similar al visto en la etapa de entrenamiento, genere archivos databatch que contengan toda la información pertinente para el correcto funcionamiento de la red neuronal. Primero se generan 3 imágenes adicionales donde cada una corresponde a la imagen original escalada en un factor de 2/3, 4/9 y 3/2. Para cada imagen se determina cual de sus dimensiones(horizontales y verticales) es mayor y a esa se le asigna un valor auxiliar fijo de 100, con una simple regla de tres se determina el valor correspondiente a la dimension menor. Utilizando estos valores, se determina un par de constantes S entendidas como el cociente de la división del valor original de la dimension por el valor auxiliar de está. En cada una de las imágenes pasamos una sliding window con dimensiones de 81x81 píxeles por toda la imagen, con el fin de generar un conjunto de particiones; el desplazamiento de esta ventada esta dado por las constantes S horizontal y vertical calculadas, la utilización de estas constantes en el desplazamiento de la ventana garantizan que, para las cuatro imágenes de diferente escala, se generen igual cantidad de elementos pero con diferentes grados de solapamiento, hecho de suma importancias para el análisis de los resultados dados por la red neuronal.

5 Resultados

6 Conclusiones