

Detección de Poses con Just Dance

Programa que por medio de una webcam sea capaz de construir el esqueleto que le de seguimiento al cuerpo humano

Roderick Aparicio	8-916-593
Isaac Bethancour	8-917-2263
Alan Castro	8-912-1890
Jesús De Gracia	8-1086-1646
Efrain Escobar	8-772-1697
Guillermo Espino	8-925-2235
Daniel González	8-907-1404
Allan Marín	EC-20-12127
Elvin Marín	EC-20-12126
David Morán	2-741-87

Lourdes Moreno	8-920-640
Kaiser Obaldia	8-898-703
Yeny Ortega	8-923-1263
Fermin Povaz	8-932-1661
Kevin Ruedas	8-910-1800
Krisbel Sanjur	2-738-644
Alexander Santana	8-933-1167
Stephanie Tejeira	8-924-2239
Victor Valenzuela	8-931-2246
Ana Villarreal	8-917-2049

Abstract- In this laboratory we will see the estimation of the human pose in the form of a skeleton with the help of Neural Networks, which allows us to represent the orientation of a person in a graphic format. Using a set of coordinates that can be connected to describe a person's pose. Each coordinate of the skeleton is known as a key point. A connection between two key points is known as a limb. This technique has applications in innumerable fields, such as for the recognition of activities, motion capture and augmented reality, training of other robots, monitoring for electronic equipment among.

Resumen

En este laboratorio veremos la estimación de la pose humana en forma de esqueleto con ayuda de las Redes Neuronales, que permite representar la orientación de una persona en un formato gráfico. Mediante un conjunto de coordenadas que se pueden conectar para describir la pose de una persona. Cada coordenada del esqueleto se

conoce como punto clave. Una conexión entre dos puntos claves se conoce como extremidad. Esta técnica tiene aplicaciones en innumerables campos, como para el reconocimiento de actividades, captura de movimiento y realidad aumentada, entrenamiento de robots, seguimiento para equipos electrónicos entre otros.

Introducción

En este caso utilizaremos Opencv y Tensor Flow con el procesamiento de redes neuronales por High Resolution Net. Se utilizará la app Just Dance Now para poder efectuar distintos bailes y estos movimientos se visualizarán a través de la webcam y con la ayuda del programa se colocara el video para que el programa le dé seguimiento a los movimientos que se generen.

Palabras Claves

Tensor Flow, OpenCV, Yolo, Coco, redes neuronales. HrNet, extremidad, estimación, puntos claves.

Objetivo

Implementar un programa que por medio de una webcam y video sea capaz de construir el esqueleto que le de seguimiento al cuerpo humano.

Requerimientos y Métodos

Librerías python que se necesita para poder realizar este laboratorio son los siguientes:

- ❖ **Matplotlib:** produce figuras de calidad en una variedad de formatos impresos y entornos interactivos en todas las plataformas.
- ❖ **Numpy:** nos ayuda a manejar arreglos de varias dimensiones para el manejo de cálculo con una alta eficiencia.
- ❖ **Opencv:** herramienta utilizada para elaborar prototipos rápidos para problemas de visión por computadora.
- ❖ **Pillow:** maneja el procesamiento de imágenes, con varios tipos de formatos compatibles.
- ❖ **Vidgear:** permite procesar, enviar y recibir cuadros de video en tiempo real.
- ❖ **Torch:** para el cálculo de redes neuronales.
- ❖ **Torchvision:** conjuntos de datos y modelos de imágenes y video para el aprendizaje profundo.
- ❖ **Tqdm:** Medidor de progreso rápido para evitar la regresión de rendimientos.
- ❖ **Tensor Flow:** para el aprendizaje automático y otros

datos cálculos con otros datos descentralizados.

- ❖ **Yolov3:** sistema utilizado para la detección de objetos en tiempo real.
- ❖ **Coco:** posee data de una variedad de objetos de la vida cotidiana para analizar.
- ❖ **HrNet:** red neuronal para la detección de objetos y clasificación de imágenes o videos.

El proyecto está pensado para ser procesado mediante la implementación de CUDA sin embargo se puede usar procesamiento en de cpu regular.

Metodología

En visión artificial con esta librería detectamos la posición y orientación de un objeto. Esto significa detectar la ubicación de puntos clave de objetos particulares.

Por ejemplo, en el problema de la estimación de la pose del rostro (es decir, detección de puntos de referencia faciales), esto significa que detectamos un rostro humano.

En este artículo trabajaremos en Estimación de la postura humana utilizando OpenCV, donde detectar y localizar las principales partes del cuerpo como articulaciones, hombros, rodillas, muñecas, etc.

Dataset:

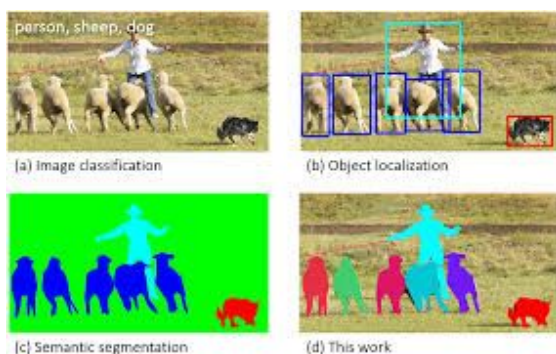
Inicializamos un nuevo objeto desde COCO Dataset. Los índices de imágenes y anotaciones se cargan y

almacenan en la memoria. Las anotaciones se pre procesan para tener una lista simple de anotaciones para iterar.



Puntos de colores COCO

Los cuadros delimitadores se pueden cargar desde la verdad básica o desde un archivo. Al cargar los datos se procede a convertir las imágenes en tensores y se normaliza su clasificación.



Ejemplo de clasificación en Coco Dataset

Al cargarse la data, se hace una comparación entre los puntos claves de la imagen con los preestablecidos del esqueleto humano de openCv.

Entrenamiento:

Creamos una clase de entrenamiento que es la que proporciona el manejo básico para la formación de la HRNet. A pesar de poderse personalizar utilizamos las funciones básicas del algoritmo, con un único método que se

llama directamente. Luego de iniciar el proceso se crea una carpeta de registro, se inicializa el modelo HRNet y se guardan los pesos y puntos de control previamente entrenados cargados. Se definen los dataloaders, la función de pérdida y el optimizador para poder evaluar la precisión y obtener predicciones sobre la entrada.

Testing:

Creamos una clase para las pruebas que proporciona una herramienta básica para probar los puntos de control de HRNet. Al definir un nuevo objeto de prueba, se inicializa el modelo HRNet y se carga el punto de control guardado.

Lo que maneja el DataLoader y la función de pérdida.

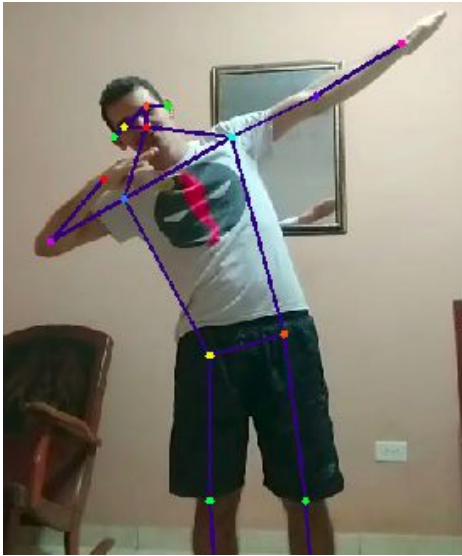
High Resolution Net:

Con esto definimos la forma de trabajo simple y personalizable para cargar la red HRNet, cargar el oficial pre-entrenado, pesos y predecir la pose humana en imágenes o videos individuales. También se incluye soporte para varias personas con el detector YOLOv3 (y está habilitado de manera predeterminada).

HRNet (y YOLOv3) se inicializan en el dispositivo del torch y sus pesos pre-entrenados se cargarán desde el disco.

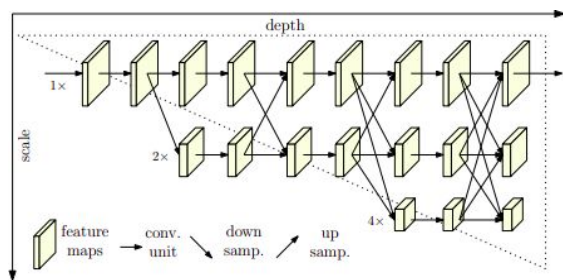
Luego de haber completado la estructura de aprendizaje, se procede a correr el programa de forma escalonada para poder conocer si los puntos claves

del esqueleto se están representando de forma correcta.



Definiendo los puntos claves del cuerpo

Interpretación del algoritmo



Representación del funcionamiento del algoritmo

Este algoritmo busca dar una mejor representación al problema de estimación de pose mediante el enfoque de aprendizaje de representaciones confiables de alta resolución. La gran mayoría de métodos que se utilizan, recuperan representaciones de alta resolución de representaciones de baja resolución producidas por una red de alta a baja resolución. En cambio, la red que se utiliza, mantiene representaciones de alta resolución durante todo el proceso. Básicamente

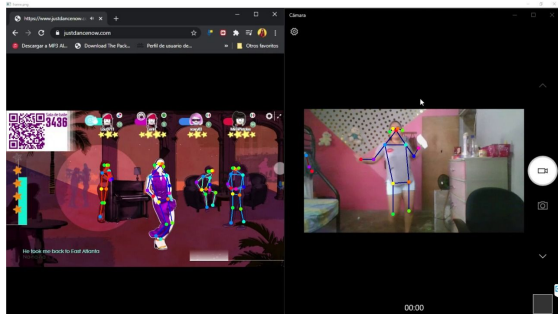
se inicia desde una subred de alta resolución como primera etapa, luego se van agregando gradualmente subredes de alta a baja resolución una a una para formar más etapas y conectamos las subredes de múltiples soluciones en paralelo. Se realizan fusiones de múltiples escalas repetidas de manera que cada una de las representaciones de alta a baja resolución recibe información de otras representaciones paralelas una y otra vez, lo que genera representaciones bastante saturadas de alta resolución. Como resultado, el mapa de calor de puntos clave previsto es potencialmente más preciso y espacialmente más preciso.

Pruebas y Resultados

Podemos apreciar algunas pruebas que realizamos con ayuda del programa desarrollado, en el cual participaron algunos compañeros en estas pruebas. el video fue grabado por pantalla y luego procesado con el sistema, para que este se pudiera procesar utilizando el opencv y este dibujara la silueta de la persona.



implementación general de los puntos en el esqueleto.



Pruebas realizadas con la app just Dance

Propuesta a Futuro

Como propuesta a futuro de este laboratorio se considera una mejor optimización y renderización del video, también la mejora del código para que se procese mejor el video y se pueda ver en tiempo real. Al hacer estas mejoras se podrá obtener una mejor imagen en la webcam, mejor procesamiento en los movimientos de tiempo real y por último un mayor enfoque en los objetos, es decir, que no se salga ningún punto para dar forma al esqueleto humano.

Conclusión

Podemos concluir que al realizar este laboratorio pudimos tener un conocimiento más amplio sobre las redes neuronales como son las redes profundas (DNN), también un nuevo conocimiento sobre los OpenCV y los Tensor Flow. Además logramos que este sistema llegará a el objetivo que deseábamos en el laboratorio de DeepPose mediante todas la técnicas y requerimientos utilizados. También comprendimos el comportamiento del algoritmo para que lograra procesar el video y se pudiera visualizar la webcam con el esqueleto humano y los

movimiento que se hacía con simultáneo con el Just Dance . En cuanto a la estimación de poses humanas se han logrado grandes avances, lo que nos permite servir mejor a la miriada de aplicaciones que son posibles con ella.

Bibliografía

- *Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields*. (2017, 25 julio). [Video]. YouTube.
https://www.youtube.com/watch?v=OgQLDEAjAZ8&feature=emb_title
- P.(2020). *Pavankunchala/Human-Pose-Estimation-OpenCV*. GitHub.
<https://github.com/Pavankunchala/Human-Pose-Estimation-OpenCV>
- C.(2020a). *code-conquer/Human-pose-estimation*. GitHub.
<https://github.com/code-conquer/Human-pose-estimation>
- Kalbande, A. (2020, 9 septiembre). *Human Pose Estimation using OpenCV*. Fireblaze AI School Blogs.
<https://www.fireblazeaischool.in/blogs/human-pose-estimation-using-opencv/>