

Bienvenidos a la clase de 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 the next article we will talk about how we will implement a program that, through a webcam, is capable of building a skeleton which must track the movements that are generated for this program. We will use the human Pose library.

Introducción

En este laboratorio veremos la estimación de la pose humana en forma de esqueleto con ayuda de las Redes Neuronales, utilizando Opencv y TensorFlow. Se utilizará la versión gratuita de la App just dance 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, para poder realizar este programa se utilizaron High Resolution Network (HRNet), Yolo Version 3, la dataset de Coco y PyTorch.

Palabras Claves

TersonFlow, OpenCV, Yolo.

Objetivo

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

Requerimientos

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:** es un paquete de python que nos ayuda a manejar arreglos multidimensionales de alta eficiencia y diseñados para cálculo científico.
- ❖ **Opencv:** herramienta adecuada para la creación rápida de

prototipos de problemas de visión por computadora

- ❖ **Pillow:** agrega las capacidades de procesamiento de imágenes, permitiendo un amplio soporte de formato de archivo
- ❖ **Vidgear:** se encarga del procesamiento de video, permite procesar, enviar y recibir cuadros de video en tiempo real.
- ❖ **Torch:** para el cálculo Tensores y redes neuronales dinámicas
- ❖ **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.
- ❖ **Tensorflow;** para el aprendizaje automático y otros datos cálculos con otros datos descentralizados.

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

Metodología

La metodología utilizada para realizar este laboratorio fue las librerías OpenCV Y Terson Flow

1.OpenCV: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.

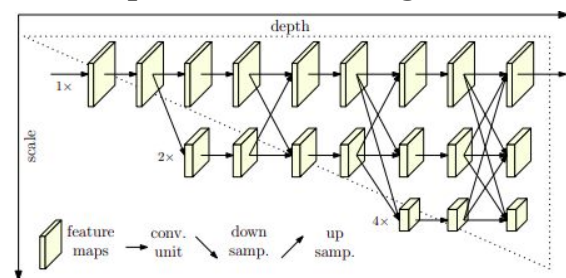
Por ejemplo, en el problema de la estimación de la pose del rostro (es decir, detección de puntos de referencia faciales), esto significa que detectamos un rostro humano.

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.



Localización de las parte del cuerpo

Comportamiento del algoritmo

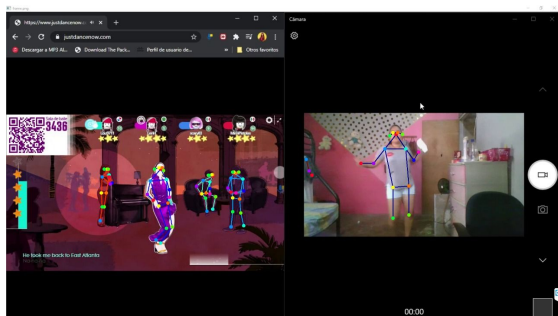


Este algoritmo busca dar una mejor representación al problema de estimación de pose mediante el enfoque de aprendizaje de representaciones confiables de alta resolución. La gran mayoría de métodos que se utilizan, recuperan representaciones de alta resolución de representaciones de baja resolución producidas 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.

Procedimientos de Pruebas

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.



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 Terson 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 .

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/>