UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

Detección y conteo de dedos para la ejecución de tareas de brazo robótico 6DOF Jetson Nano.

John Alexander Hidalgo Abril

Ingeniería Electrónica

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Ingeniero Electrónico

Quito, 19 de diciembre de 2023

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Colegio de Ciencias e Ingenierías

HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA

Detección y conteo de dedos para la ejecución de tareas de brazo robótico 6DOF Jetson Nano.

John Alexander Hidalgo Abril

Nombre del profesor, Título académico **Diego Benítez, Ph.D.**

Quito, 19 de diciembre de 2023

© DERECHOS DE AUTOR

Nombres y apellidos: John Alexander Hidalgo Abril

Código: 00206606

Cédula de identidad: 1850082742

Lugar y fecha: Quito, 19 de diciembre de 2023

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

RESUMEN

Este artículo presenta el desarrollo de un sistema que opera un brazo robótico para la manipulación de objetos basado en las posiciones de los dedos levantados de la mano humana detectada por la cámara del robot, que demuestra el reconocimiento de cuantos dedos están levantados en tiempo real para la interacción robot-humano. Para lograr esto, se desarrolló una red neuronal convolucional capaz de identificar 21 puntos de interés de la mano detectada en tiempo real. La operación del brazo robótico se implementó mediante una computadora Jetson Nano, una cámara web y librerías de Python como son OpenCV, TensorFlow e I2Ctools. Se utilizo un conjunto de 30k fotos de manos en distintas posiciones para generar el modelo de detección de dedos que demostró una precisión del 92.7% y un error del 7.3% durante el entrenamiento y validación. En tanto que en las pruebas en tiempo real del prototipo se obtuvo una precisión de prueba de 93.8\% con un error de 6.2\%, esta prueba de concepto demuestra que en el futuro próximo se puede realizar aplicaciones avanzadas que aprovechen los gestos de las manos que puede ser construidas por el usuario.

Palabras clave: redes neuronales convolucionales, control de brazo robótico, reconocimiento de mano.

ABSTRACT

This article presents the development of a system that operates a robotic arm for the manipulation of objects based on the positions of the raised fingers of the human hand detected by the robot's camera, which demonstrates the recognition of how many fingers are raised in real time for robot-human interaction. To achieve this, a convolutional neural network was developed capable of identifying 21 points of interest of the hand detected in real time. The operation of the robotic arm was implemented using a Jetson Nano computer, a webcam and Python libraries such as OpenCV, TensorFlow and I2Ctools. A set of 30k photos of hands in different positions was used to generate the finger detection model that demonstrated an accuracy of 92.7% and an error of 7.3% during training and validation. While in the real-time tests of the prototype a test accuracy of 93.8\% was obtained with an error of 6.2\%, this proof of concept shows that in the near future advanced applications can be made that take advantage of the hand gestures that can be built by the user.

Key words: Convolutional neural networks, robotic arm control, hand recognition.

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

El progreso tecnológico en la era contemporánea se enfoca en la interacción cada vez más estrecha entre humanos y máquinas. Estos dispositivos tienen la finalidad de simplificar procedimientos y garantizar la seguridad humana. La complejidad de una máquina se determina por la sofisticación de las tareas que puede llevar a cabo. Con el avance tecnológico, las máquinas pueden ahora comunicarse de forma más efectiva con los humanos, identificando sus necesidades y proporcionando interacciones más enriquecedoras. Las aplicaciones de esta tecnología abarcan desde máquinas que responden a comandos de voz hasta robots autónomos. La inteligencia artificial (IA) posibilita este avance al permitir que las máquinas imiten el comportamiento humano [1]. Se anticipa que, en el futuro, los robots humanoides coexistirán con los humanos como compañeros en diversas situaciones, ya sea en el ámbito personal o laboral [2].

El reconocimiento de gestos y movimientos de la mano a través de redes neuronales ha ganado atención debido a su capacidad para interpretar de manera precisa y rápida la información visual [1]. La utilización de una CNN específicamente diseñada para la identificación de posiciones de dedos constituye un paso crucial hacia la implementación de sistemas de control más intuitivos y adaptativos. Además, considerando que no existe un único método para la detección de manos y el reconocimiento de las posiciones de los dedos, se escogió una solución dependiendo del dominio de la aplicación a realizarse, el entorno a utilizarse e incluso basándose en el usuario que utilizara la aplicación como menciona en [2].

Existen distintos tipos de detección y reconocimiento de la mano que son comparados en [2]-[7]. Estos hacen alusión a enfoques que clasifican las posiciones de la mano en base al color, la forma, la apariencia y el movimiento de la mano para de manera combinada poder arrojar un resultado deseado. En nuestro caso de estudio, se aplica

análisis de profundidad y apariencia mediante entrenamiento automático con MediaPipe que permite la detección y reconocimiento de la mano para la obtención de 21 puntos de interés de la mano [4]. Esto permite ser aplicado en distintos ámbitos como el de nuestros intereses de la manipulación y toma de decisiones del brazo robótico.

Este trabajo aborda la importancia de la detección precisa de posiciones de los dedos y su relación directa con la manipulación efectiva de un brazo robótico. Se explorarán en detalle los avances más recientes en el campo, analizando investigaciones previas y destacando los métodos y enfoques más relevantes. Asimismo, se presentará una metodología específica basada en CNN, detallando su arquitectura y entrenamiento para lograr una detección robusta y en tiempo real ejecutado por el Jetson Nano incluido en el brazo robótico, además de usar herramientas como es OpenCV y MediaPipe para el procesamiento de las imágenes adquiridas en tiempo real.

MATERIALES Y METODOLOGÍA

A. Hardware.

El prototipo realizado este compuesto por distintos dispositivos electrónicos que se encuentran en la Fig.1. El principal dispositivo con el que se trabajó es el brazo robótico de 6 ejes, específicamente el brazo robótico AI Vision Yahboom DOFBOT [8]. Este brazo este hecho de una aleación de aluminio que proporciona fuerza sin perjudicar el peso al realizar movimientos. Por otro lado, el control y manipulación de los servomotores del brazo robótico son ejecutados por una placa de expansión conectada mediante I2C a una tarjeta de desarrollo Jetson Nano de Nvidia[9]. Esta tarjeta de desarrollo mediante el Sistema Operativo Robótico [10](ROS por sus siglas en Inglés) controla las acciones realizadas por cada servomotor dado su facilidad de código abierto. Por último, se hace uso de una cámara Logitech Brio 4K Webcam [11] que permite la corrección de color y mayor campo de visión para poder ejecutar un correcto procesamiento de las imágenes captadas en tiempo real que son procesadas mediante el algoritmo aplicado con OpenCV [12].

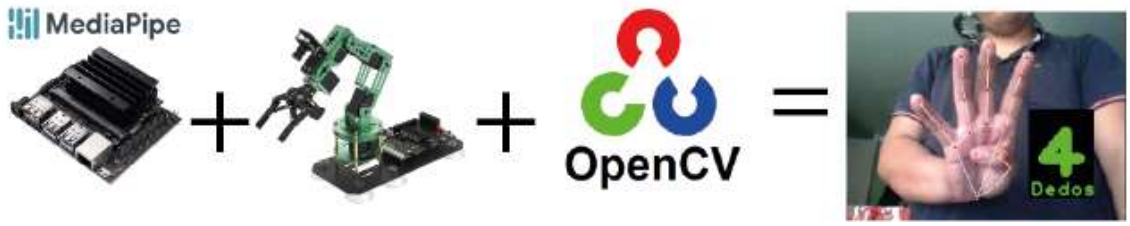

Figure 1: Prototipo del proyecto.

B. Base de datos y Detector de manos.

Para el entrenamiento de la CNN se utilizó un dataset de keras Finger Hand [13], el cual dispone de un conjunto de 200k imágenes de manos en distintas posiciones los dedos. Sin embargo, se utilizó un total de 26k imágenes para entrenamiento y 4K para la

verificación de el correcto funcionamiento de la red neuronal. En la Fig.2 se encuentra un ejemplo de las imágenes utilizadas para tener un resultado optimo del funcionamiento de la CNN.

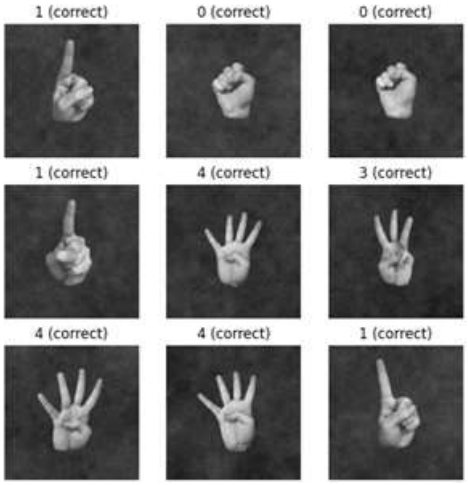

Figure 2: Ejemplos de base de datos de entrenamiento.

La detección de manos se resuelve mediante el algoritmo MediaPipe Hand Landmarker [14], este algoritmo se basa en un modelo de aprendizaje automático que mide datos de imágenes de flujo continuo que permite la detección de la palma y generación de puntos de interés de cada dedo para así mantener en tiempo real el correcto conteo de dedos. Esto se debe a que el detector de los puntos de intereses está calculando las coordenadas dentro de la imagen donde se ha detectado la mano. Ya que se toma un cuadro de video como entrada el algoritmo realiza compensaciones de iluminación, cambio de canal y normalización para generar un resultado similar a los puntos detectados en la Fig.3. Y mediante la votación y las predicciones realizadas por el modelo desarrollado arroja el valor de dedos contados al compararlos con el algoritmo de detección de dedos levantados.

Figure 3:Detector de dedos con puntos de interés.

C. Modelo de Aprendizaje Profundo.

La arquitectura de la red neuronal propuesta se muestra en la Fig.4, esta fue implementada mediante las librerías de TensorFLow y Python en una computadora virtual de Google Collab. El modelo consta de seis capas de convolución 2D, cada capa tiene seguido un operador de agrupación máxima. En la Fig.3, se muestra los tamaños de los núcleos de convolución, el volumen de cada capa y su operador de agrupación correspondiente. A la salida de la sexta capa de convolución se le conecta como entrada a una red complementaria de 6 capas, las cuales tienen 512 neuronas ocultas, y son aplicadas la función de activación excepto la última capa que se toma como la salida. La última capa consta de 6 neuronas, una para cada posición de los dedos levantados generando la salida deseada. Por último, la red neuronal presenta normalización por lotes [13] para evitar el sobreajuste y poder tener seis posibles salidas en función de los rasgos de las imágenes con las que se ha entrenado.

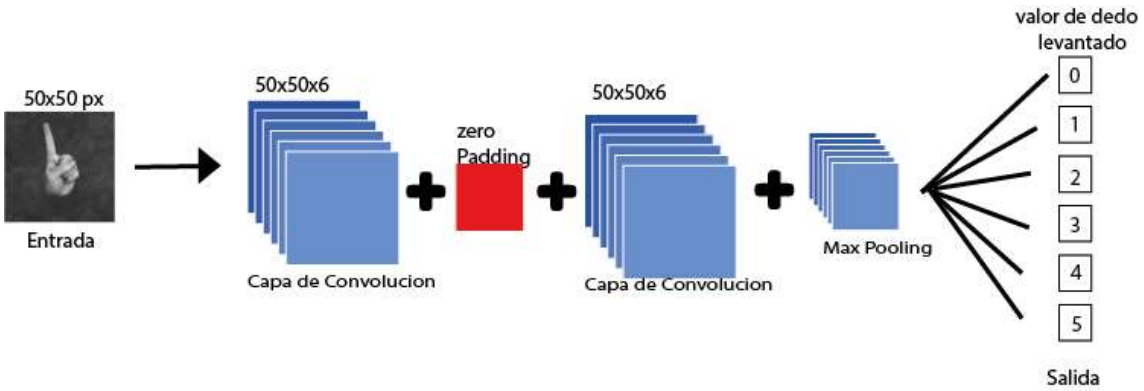

Figure 4: Estructura de la red neuronal convolucional.

D. Control de Brazo Robótico.

El brazo robótico Dofbot se controla mediante un conjunto de comandos ROS precargados. El movimiento del brazo depende del número de dedos levantados detectados, por lo que se definen varias posiciones en los seis servos. Para este propósito, se utilizan tres comandos básicos. El primer comando cambia los ángulos de los seis servos y controla su velocidad. El segundo comando permite seleccionar un servo, cambiar su ángulo y controlar su velocidad. El tercer comando permite que el brazo llegue a una posición específica antes de pasar a la siguiente. Del mismo modo, se definen siete grupos de acciones para controlar los movimientos del brazo: el primero mueve la posición del brazo para enfocar la cámara en la persona para su detección, mientras que los grupos del dos al siete permiten al robot recoger y mover un cubo preestablecido en un lugar específico dentro del rango de trabajo del brazo robótico.

Para detectar la mano de una persona, el brazo se mueve inicialmente a la primera ubicación usando los comandos del primer grupo. Cuando se detecta una mano, el brazo se mueve dependiendo de los dedos levantados al cuadrante donde está ubicado el cubo que corresponde a esa sección. Sin embargo, durante las pruebas, se observó un ligero movimiento del sujeto durante la predicción que afectó a la clasificación, ya que el sistema solo analiza el valor de pronóstico más reciente. Por lo tanto, se implementó un sistema de votación para hacer un movimiento. Se realizan diez predicciones en tiempo real eliminando la posibilidad de un valor intermedio se toma el valor con mayor frecuencia que apareció en la etapa de detección y cada que sale del contador, este se reinicia y el robot ejecuta la directriz asignada.

EXPERIMENTO Y RESULTADOS.

A. Configuración Experimental.

1.) Configuración del Modelo.

El modelo fue diseñado para recibir imágenes en escala de grises de 50 pixeles de ancho y alto, y clases de salida de cero, uno, dos, tres, cuatro, y cinco respectivamente para los dedos que pueden ser alzados al ser detectada una mano. Debido a que se necesita un robusto procesador para la detección y diferenciación de manos izquierda y derecha se consideró el entrenamiento para la detección de la mano derecha y afianzar los datos de detección en tiempo real mediante Mediapipe [14]. Por otro lado, se utilizó el algoritmo de descenso de gradiente estocástico para entrenar al modelo con una tasa de aprendizaje del 0.01 y una tasa de decaimiento establecida por la tasa de aprendizaje dividida por el número de épocas. Estas épocas se establecieron entre 60 y 100. Además, al finalizar cada época se realizó el cálculo de la perdida y la precisión que demuestra que los valores alcanzados son similares en el entrenamiento como en la validación de la red. Como resultado se indica que el modelo no presento sobreajuste logrando obtener una red neuronal con una pérdida del 7.3% y una precisión del 92.7% capaz de ser utilizada en tiempo real.

2.) Procesamiento de Imágenes.

En el estudio se utilizó 4000 imágenes con un tamaño de 50x50 pixeles aproximadamente para cada uno de los gestos de la mano para poder clasificar los gestos en función de los dedos levantados: 0 dedos, 1 dedo ,2 dedos hasta 5 dedos levantados. Para garantizar que las imágenes de cada gesto se la mano con 0,1,2,3,4 o 5 dedos levantados se intercalaran, las imágenes se mezclaron y normalizaron al azar. Además, se realizó un aumento de datos para aumentar la eficiencia al permitir que el modelo recibiera nuevas variaciones de las imágenes en cada época durante el proceso de entrenamiento. Transformaciones aleatorias como el desplazamiento de la imagen en ancho y altura, el recorte de partes de la imagen, el zoom a un rango de 0.3 volteos

horizontales y el llenado de nuevos píxeles usando el píxel más cercano, el recorte o el desplazamiento, se aplicaron rotaciones en un ángulo de 20 ° a las imágenes en el conjunto de entrenamiento.

3.) Pruebas en Tiempo Real.

El módulo, el modelo entrenado y el control del brazo robótico se ensamblaron en un prototipo de tiempo real, el cual fue utilizado en la ciudad de Ambato para que diez participantes al azar realizaran el experimento en tiempo real. Como se muestra en la Fig. 5, el robot se colocó sobre una mesa (90cm desde el suelo). Se explico el funcionamiento del prototipo a los participantes y se instruyó a los mismo a mostrar la mano derecha al robot. Durante cinco oportunidades, cada participante modifico la posición y el número de dedos levantados para que el robot opere dependiendo de la posición que obtuvo. Ante cualquier inconveniente con la detección se indicó a los participantes de retirarse cualquier objeto cercano a la mano y el robot realizó las correctas ordenes tomadas por el número de dedos levantados por el participante. Durante estas pruebas, existió un seguimiento del rendimiento del modelo utilizando métricas de precisión y error.

Figure 5: Experimento en tiempo real.

4.) Métricas de Evaluación.

El rendimiento del modelo se monitoreo utilizando las métricas de error durante las fases de entrenamiento y pruebas en tiempo real. Además, se utilizó la precisión (ACC)

[15] para poder evaluar las predicciones en el prototipo en tiempo real afianzando los resultados arrojados por el modelo en funcionamiento y así poder desarrollar nuevas ideas a partir del proyecto elaborado.

B. Resultados.

El modelo fue entrenado y probado mediante el conjunto de datos de las posiciones de la mano, logrando una precisión del 93.8% y un error del 6.2% como se muestra en la Fig.6. Este gráfico muestra las cuatro curvas, con las funciones superiores que demuestran la precisión del entrenamiento y validación mientras que las funciones inferiores ilustran las pérdidas del entrenamiento y validación. Estos resultados muestran que si la curva de validación tiende a 1 indica que el modelo es funcional y si el error tiende a 0 y se mantiene estable indica que el modelo puede ser utilizado en pruebas de tiempo real. El prototipo final realizado funcionó acorde a los valores arrojados por el entrenamiento, los datos adquiridos en el prototipo son mostrados en la tabla 1 y los porcentajes de precisión se encuentran en la tabla 2. Los resultados fueron consistentes y los errores ligeramente más altos se dieron al aumentar los dedos levantados.

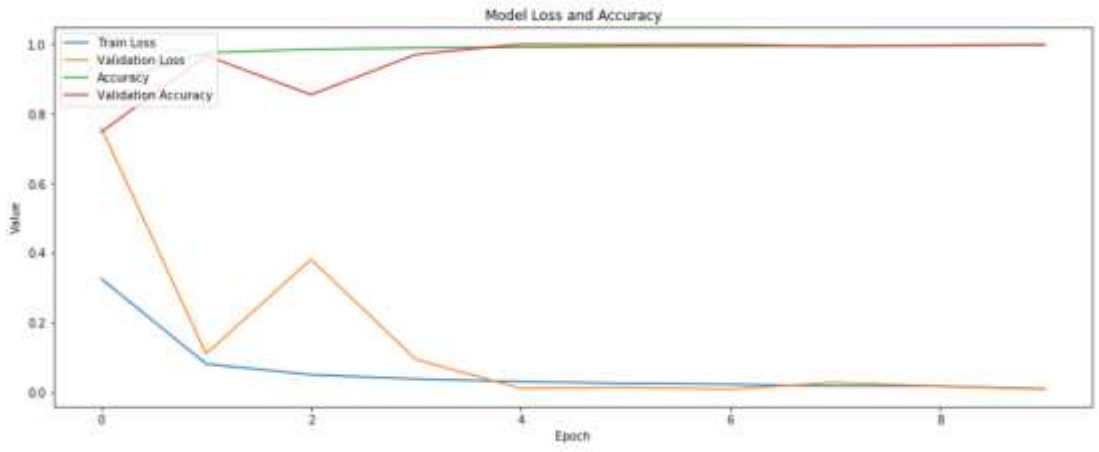

Figure 6: Curva de entrenamiento de la red neuronal.

Predicción							
Valor Real		0	1	2	3	4	5
	0	**49**	0	0	0	0	0
	1	0	**47**	1	2	1	1
	2	0	1	**48**	0	0	0
	3	0	1	1	**47**	2	1
	4	1	1	0	1	**46**	3
	5	0	0	0	0	0	**45**

Tabla 1:Matriz de Verdad de la red neuronal.

Resultado	Entrenamiento	Valor Real					
		0	1	2	3	4	5
Presición (%)	94	98	94	96	94	92	90
Error (%)	6	2	6	4	6	9	10

Tabla 2: Rendimiento del modelo de detección de dedos.

CONCLUSIONES Y TRABAJOS FUTUROS

Esta investigación demuestra la viabilidad de usar un modelo de detección de dedos alzados de la mano basados en una red neuronal convolucional para operar un brazo robótico y entregar un objeto de acuerdo con el número de dedos levantados de la mano de un individuo frente al ´área de reconocimiento del robot. El prototipo en tiempo real del sistema funciono de manera efectiva y produjo un nivel aceptable de precisión. Sin embargo, el movimiento actual del brazo está controlado por la posición, por lo que, si el objeto de entrega no está en un lugar predeterminado, el brazo no lo entregará. Para mejorar el sistema, el trabajo futuro tiene como objetivo incluir el lenguaje de señas y permitir que el brazo robótico explore su rango de movimiento y visión para buscar objetos particulares.

.

REFERENCIAS BIBLIOGRÁFICAS

[1] N. C. Kiliboz and U. Gudukbay, “A hand gesture recognition technique for human-computer interaction,” Journal of Visual Communication andImage Representation, vol. 28, pp. 97–104, 2015.

[2] Juan Pablo Wachs, Mathias K ¨olsch, Helman Stern, and Yael Edan, “Vision-Based Hand-Gesture Applications,” Communications of the ACM, vol. 54, n.o2, pp. 60-71, febrero 2011.

[3] X. Zabulis, H. Baltzakist, and A. Argyros, “Vision-based Hand Gesture Recognition for Human-Computer Interaction,” in The Universal Access Handbook (Human Factors and Ergonomics), Constantine Stephanidis,Ed.: CRC Press, 2009, ch. 34, pp. 34.1-34.30.

[4] Defferrard, M., Bresson, X., y Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. CoRR, abs/1606.09375.

[5] Mathias K ¨olsch and Matthew Turk, “Robust Hand Detection,” 6th IEEE International Conference on Automatic Face and Gesture Recognition (FGR’04), pp. 614-619, 2004.

[6] Hussain, S., Saxena, R., Han, X., Khan, J. A., & Shin, H. (2017). Hand gesture recognition using deep learning. 2017 International SoC Design Conference (ISOCC). oi:10.1109/isocc.2017.8368821

[7] Mitra, S., & Acharya, T. (2007). Gesture Recognition: A Survey. IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), 37(3), 311–324. doi:10.1109/tsmcc.2007.893280

[8] Yahboom, “Dofbot AI vision robotic arm.” [Online]. Disponible: http://www.yahboom.net/study/Dofbot-Jetson\ nano

[9] Nvidia, “Jetson nano 2gb developer kit,” Oct 2020. [Online]. Disponible: https://developer.nvidia.com/embedded/jetson-nano-2gb-developer-kit

[10] A. Koubaa, Robot Operating System (ROS): The Complete Reference (Volume 7). Springer Nature, 2023, vol. 1051.

[11] Logitech,“Logitech brio 4k webcam.”[Online].Disponible:https://www.logitech.com/en-us/products/webcams/brio4k-hdrwebcam.960-001105.html

[12]J.Howse and J. Minichino, Learning OpenCV 4 Computer Vision with Python 3: Get to grips with tools, techniques, and algorithms for computer vision and machine learning. Packt Publishing Ltd, 2020.

[13] F. Zhan,” Hand Gesture Recognition with Convolution Neural Networks,” 2019 IEEE 20th International Conference on Information Reuse and Integration for Data

Science (IRI), Los Angeles, CA, USA, 2019, pp. 295-298, doi: 10.1109/IRI.2019.00054.

[14] S. S. Teja Gontumukkala, Y. Sai Varun Godavarthi, B. R. Ravi Teja Gonugunta and S. Palaniswamy," Hand Cricket Game using CNN and MediaPipe," 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2022, pp. 1-6, doi:10.1109/ICCCNT54827.2022.9984411.

[15] W. Wu, M. Shi, T. Wu, D. Zhao, S. Zhang and J. Li," Real-time Hand Gesture Recognition Based on Deep Learning in Complex Environments," 2019 Chinese Control And Decision Conference (CCDC), Nanchang, China, 2019, pp. 5950-5955, doi: 10.1109/CCDC.2019.8833328.