# Investigating Hand Gesture Recognition Models: 2D CNNs vs. Visual Transformers

Vasyl Teslyuk[1], Volodymyr Chornenkyi[1], Volodymyr Tsapiv[1] and Iryna Kazymyra[1]
[1] *Lviv Polytechnic National University, 12 S. Bandera Str., Lviv, Ukraine*

**Abstract**
This paper presents an investigation into hand gesture recognition aimed at enhancing military training, improving human-machine interactions, and facilitating communication for individuals with disabilities. Through a comprehensive analysis, it explores both established computer vision methodologies and contemporary deep learning trends. The study examines the effectiveness of models utilizing 2D convolutional neural networks (2D-CNNs) and visual transformers (HGR-ViT). The research evaluates their performance by deploying models trained on ASL and NUS-II datasets, encompassing diverse sign language images, utilizing various performance metrics such as recall, precision, and the F1 score. The analysis identifies scenarios where 2D-CNNs and visual transformers achieve superior accuracy while acknowledging constraints influenced by environmental variables and computational resources. This work contributes to advancing hand gesture recognition, particularly in contexts of military training and accessibility, offering insights into cutting-edge deep learning architectural paradigms.

**Keywords:**
Deep learning, human-machine interactions, neural networks performance, sign language datasets.

## 1. Introduction

The issue of hand gesture recognition occupies a central position in academic research in the fields of computer vision and deep learning. The anticipated development of virtual (VR) and augmented reality (AR) technologies, where gestures become a significant means of interaction, increases the relevance and necessity for further research. Primarily, gesture recognition systems have the potential to revolutionize military training, human-machine communication, and particularly communication methods for individuals with disabilities.

A systematic review of the scientific literature confirms the existence of numerous methods and approaches proposed to address this pressing issue. From traditional image processing methods to modern deep learning models, many proposals have significantly contributed to improving the accuracy and efficiency of gesture recognition. However, significant challenges remain, including real-time operation, adaptation to changing conditions, and integration of various sensors.

This study focuses on innovative approaches to hand gesture recognition, particularly the analysis of 2D convolutional neural networks and transformers. By exploring the prospects of these technologies and their combined use with modern sensors, we aim to contribute to developing reliable, efficient, and adaptive gesture recognition systems.

The main goal of this study is to investigate and compare the effectiveness of applying 2D convolutional neural networks and visual transformers in hand gesture recognition tasks.

The following research tasks were identified to achieve the stated objective:

1. Analyze modern approaches to hand gesture recognition using 2D convolutional neural networks and visual transformers.
2. Evaluate the accuracy of hand gesture recognition for 2D-CNN and ViT architectures.

[1]COLINS-2024: 8th International Conference on Computational Linguistics and Intelligent Systems, April 12–13, 2024, Lviv, Ukraine

vasyl.m.teslyuk@lpnu.ua (V. Teslyuk); volodymyr.y.chornenkyi@lpnu.ua (V. Chornenkyi); volodymyr.tsapiv.mknus.2023@lpnu.ua (V. Tsapiv); iryna.y.kazymyra@lpnu.ua (I. Kazymyra)
0000-0002-5974-9310 (V. Teslyuk); 0009-0000-0569-6623 (V. Chornenkyi); 0009-0008-2420-7483 (V. Tsapiv); 0000-0003-1597-5647 (I. Kazymyra)

3. Investigate the effectiveness of models using each of the architectures on ASL [15] and NUS-II [16] datasets.

## 2. Related Works

At the early stages of computer vision development for gesture recognition, classical methods were employed. These methods rely on manual feature extraction, such as color histograms, contours, and texture characteristics. Techniques like Canny Edge Detection, Hough Transform, Histogram of Oriented Gradients (HOG), and SIFT used to be popular in computer vision for object detection and recognition [3, 5, 7]. While effective for a limited set of tasks, these approaches cannot adapt well to various changing conditions, such as lighting variations, changes in perspective, or object shading. Additionally, manual feature extraction is time-consuming and usually does not adapt adequately to new, unexpected circumstances.

With the introduction of deep learning into gesture recognition, there has been a significant paradigm shift. Recent research utilizing deep convolutional neural networks (CNNs) with video sequences has dramatically improved the accuracy of dynamic hand gestures [1, 2, 7] and action recognition [3, 5, 6]. CNNs are also helpful in combining multimodal data [2, 7], a proven technique for gesture recognition in complex lighting conditions [2, 4]. However, real-time dynamic hand gesture recognition systems pose numerous unresolved challenges. For instance, these systems receive continuous streams of unprocessed visual data, where gestures from known classes must be simultaneously detected and classified. Previous studies, such as [2, 7, 4], consider gesture segmentation and classification separately. Two classifiers, one determining whether a gesture occurred and the other characterizing the type of gesture, were trained separately, leading to limitations in system accuracy in data streams.

One innovative approach in this area is the CNN-SPP architecture [8], which uses Spatial Pyramid Pooling to capture more extensive spatial information. This allowed the network to better adapt to different sizes and shapes of objects in the image. Another approach, based on the DenseNet architecture, was modified as EDenseNet [9], providing better generalization and object recognition. In recent research [10], a method using a regular RGB camera to determine 21 key points on the hand was proposed. For this purpose, a network was developed and trained to identify these key points. At the core of this network lies the PointNet architecture, optimized for efficient operation directly on CPUs.

Initially developed for machine translation, transformers were later recognized as a revolutionary technology in natural language processing (NLP) [11, 12]. Their unique ability to consider large temporal contexts makes them particularly effective for analyzing structural and relational information in language.

Building on this success, numerous attempts have been made to adapt transformers for computer vision tasks [13]. The Vision Transformers (ViT) model [15, 17] has drawn particular attention. Unlike traditional computer vision models that use convolutions, ViT is entirely based on transformer architecture. It aligns with NLP and computer vision approaches and demonstrates impressive results, especially when working with large datasets.

The article [18] introduced the Vision Transformer (ViT) model, which demonstrated impressive performance on image classification benchmarks by directly applying self-attention mechanisms to image patches. This groundbreaking work paved the way for subsequent research into transformer-based approaches for visual recognition tasks.

Building upon the success of ViT, recent studies have extended visual transformers to tasks beyond image classification. The authors of [14] proposed the Detection Transformer (DETR), a transformer-based architecture for object detection. By replacing conventional convolutional layers with self-attention mechanisms, DETR achieved competitive results on object detection benchmarks while offering advantages in terms of flexibility and scalability. In [19] the comparative characterization of known CNN models for object recognition was carried out. The approaches used for image preprocessing are also actively researched and developed, e.g. filtering methods and skeletonization, were considered in [20-21].

Moreover, research efforts have focused on adapting visual transformers to video-based tasks. [22] introduced the Video Transformer (ViT), a model capable of capturing spatial and temporal information in video sequences using self-attention mechanisms. ViT demonstrated promising results on action recognition benchmarks, highlighting the potential of visual transformers in video understanding tasks.

# 3. Methods

This section will explore two distinct approaches for hand gesture recognition: 2D Convolutional Neural Networks (2D CNN) and HGR-ViT (Hand Gesture Recognition Visual Transformer).

## 3.1. 2D Convolutional Neural Networks (2D CNN)

2D CNNs are adept at learning spatial hierarchies of features in images, making them well-suited for object recognition, scene understanding, and hand gesture recognition. In hand gesture recognition, input data typically consists of sequential frames capturing hand movements, each representing a 2D image.

In mathematical terms, the forward pass of a 2D CNN can be represented as follows:

Given an input $V_{in}$, of dimensions $C \times H \times W$, where $H$ and $W$ represent the height and width of the volume, respectively, and $C$ denotes the number of channels, the output $V_{out}$ is computed by convolving $V_{in}$ with a set of 2D filters $W$ of dimensions $C_f \times H_f \times W_f$, where $H_f$, $W_f$, and $C_f$ represent the height, width, and number of filters, respectively. The convolution operation is followed by an activation function $\phi$ and optional pooling operations to reduce spatial dimensions. The output volume $V_{out}$ is then passed through fully connected layers for classification.

The value of position (x, y, z) on the $j$th feature map in the $i$th layer is given by:

$$v_{ij}^{xy} = \phi\left(b_{ij} + \sum_{m}\sum_{q=0}^{H_f}\sum_{r=0}^{W_f} w_{ijm}^{qr} v_{(i-1)m}^{(x+q)(y+r)}\right), \quad (1)$$

where $\phi$ is an activation function such as tanh, RELU, or any other non-linear differentiable function, $b_{ij}$ is a bias term, $w_{ijm}^{qr}$ is the (q, r)-th value of the kernel connected to the $m$-th feature map in the previous layer.

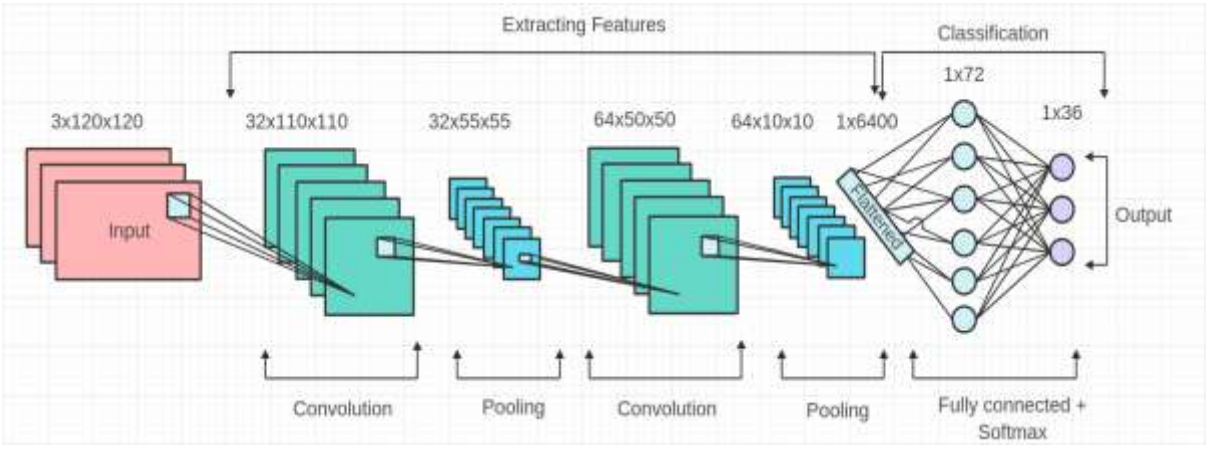


**Figure 1**: 2D CNN HGR architecture

Input data consists of single-frame images capturing hand gestures, each sized 120×120 pixels. The 2D CNN comprises convolutional layers responsible for feature extraction. Two convolutional layers are employed, each utilizing sizes 11×11 and 5×5 filters, respectively. These

filters convolve over input images to detect spatial patterns and extract relevant features associated with hand gestures.

Following the convolutional layers, two pooling layers with 2×2 kernels are added. Pooling layers serve to downsample feature maps, reducing spatial dimensions while preserving essential information. By aggregating features, pooling layers enhance computational efficiency and prevent overfitting.

The output of the convolutional and pooling layers is fed into a fully connected layer. Here, all activation values from the previous layers are combined and flattened into a vectorized representation with 6400 components. This dense layer facilitates feature aggregation and prepares the network for classification.

The Softmax layer, the final component of the network, contains output elements representing action classes. Softmax activation function normalizes the output probabilities, producing a probability distribution over the classes. This allows for probabilistic interpretation and facilitates inference of hand gestures.

### 3.2. HGR-ViT

HGR-ViT is an architecture representing Vision Transformer models designed for gesture recognition. In the initial stage, input images are normalized to a uniform size. Subsequently, these images are divided into separate patches, which are treated as sequences of pixel values. A linear projection layer, which can be learned, is applied to transform these sequences into a lower-dimensional space.

Each image patch receives additional positional encoding to retain spatial context. Then, these patches are processed using Transformer encoders, allowing the model to analyze interactions between patches. The final stage involves passing the data through a linear projection layer and a Softmax activation function to determine the probabilities of belonging to specific classes. The model training process is based on labeled data and a loss function.

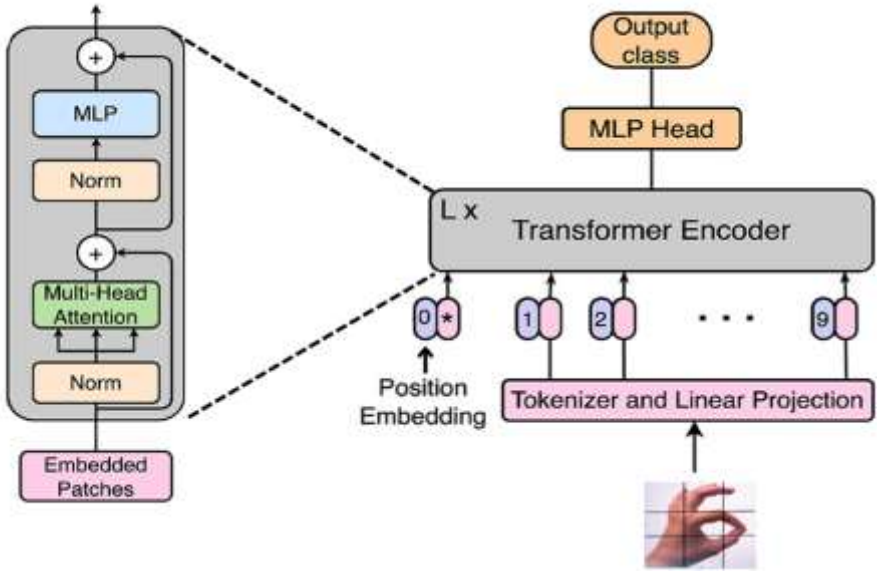


**Figure 2**: HGR-ViT architecture

To adapt hand gesture images to be segmented into 32×32 segments, they are initially resized to 256×256. Using high-resolution images with the same segment size increases the effective sequence length, improving performance. After scaling, the hand gesture images are divided into segments of standard size.

They undergo a linear projection process before passing the segments to the Transformer encoder blocks. To represent the output classification result, a learned class embedding, similar to the class token in Bidirectional Encoder Representations from Transformers (BERT)[11], is

prepended to the beginning of the sequence of embedded image segments. The following equation can represent the output of the linear projection process:

$$z_0 = \left[x_{class}; x_p^1 E; x_p^2 E; \ldots; x_p^n E\right] + E_{pos}, \quad (2)$$

where $x_{class}$ is the learned class embedding, $E$ is the learned embedding matrix, and $E_{pos}$ is the one-dimensional spatial embedding.

Segment embeddings serve as input data for the Transformer encoder, allowing ViT to detect global patterns and dependencies in the image while retaining some spatial information through segments.

The Transformer encoder is a crucial part of the Transformer model. It consists of $L$ layers, each containing two sub-layers: the multi-head self-attention (MSA) layer and the position-wise feedforward layer, also known as the multi-layer perceptron (MLP). These sub-layers are arranged sequentially, where each layer's output serves as the next layer's input, as shown in Fig. 2.

At each layer $l$, the input sequence from the previous layer is normalized using layer normalization (LN), which independently normalizes inputs across dimensions for each example. This enhances the stability of the model's representation and overall performance. The output of LN is passed through the MSA layer, and the resulting sequence is again normalized using LN. Finally, the output of the second LN passes through the MLP layer, which produces a set of updated segment embeddings.

Residual connections are added to the MLP layer to address the issue of vanishing gradients, allowing the model to learn residual functions. The equations can represent the process flow in the Transformer encoder block:

$$z`_l = MSA\left(LN(z_{l-1})\right) + z_{l-1}, \quad (3)$$

$$z_l = MLP\left(LN\left(z`_l\right)\right) + z`_l, \quad (4)$$

where $l = \overline{1, L}$ is the layer index, $z_{l-1}$ is the input sequence from the previous layer, MSA is the multi-head self-attention layer, LN is the normalization layer.

The Transformer encoder utilizes self-attention mechanisms to detect global dependencies among input tokens and multi-layer perceptrons to process the obtained representations. Residual connections and layer normalization ensure effective training and improved performance.

During training, the Rectified Adam optimizer and categorical cross-entropy loss function are used, represented by the equation:

$$L_{CE} = -\sum_{i=1}^{N} T_i \log\left(S_i\right) \quad (5)$$

where $S$ represents the Softmax probabilities and $T$ represents the labels. Early stopping and adaptive learning rate methods are also employed to prevent overfitting and improve model performance during training.

## 4. Experiment

In this study, two datasets were employed to investigate the performance of the proposed models: the American Sign Language (ASL) dataset with numbers and the National University of Singapore (NUS) dataset.

The ASL dataset with numbers, as described in [15], encompasses 36 classes of hand gestures, encompassing letters from A to Z and numbers from 0 to 9. Before model training, the images were preprocessed by resizing them to a uniform size and normalizing them. It comprises 2515 samples characterized by variations, each signed by five distinct performers. The first and second performers exhibited each symbol 25 times, except for the letter T, which contains 20 samples.

Meanwhile, the third and fourth performers demonstrated the symbols five times each, and the last performer showcased each symbol ten times. Exemplary samples for each class are depicted in Fig. 3.


**Figure 3**: Samples from ASL dataset

On the other hand, the NUS-II hand gesture dataset, introduced in [16], comprises 2000 images. This dataset incorporates 40 performers, each demonstrating gestures five times for every class, resulting in a cumulative total of 200 samples per class, thereby enhancing the diversity of hand gestures. Sample images representing each class in the dataset are illustrated in Fig. 4. Before model training, the images underwent preprocessing steps, including resizing and normalization.

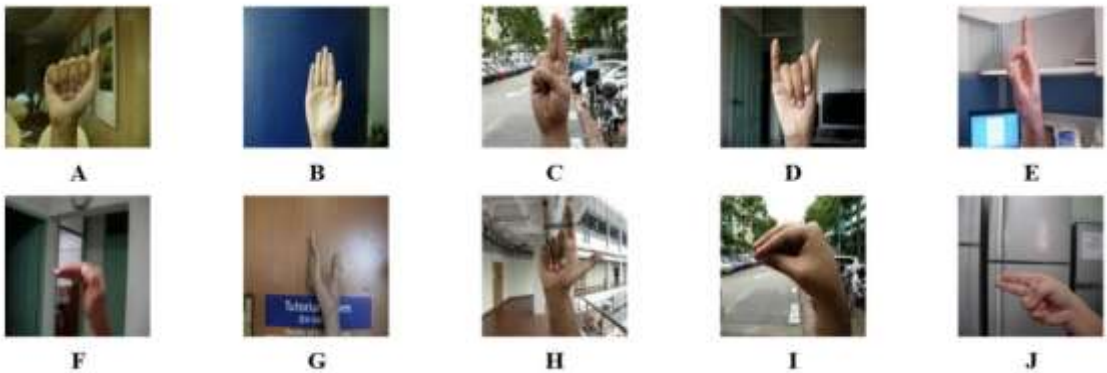


**Figure 4**: Samples from NUS-II dataset

The experiments were conducted on the system with specs listed below.

Python 3.11 was used with OpenCV 3.3 and TensorFlow on macOS Ventura, running on a notebook equipped with an Apple M1 Pro processor and 16 GB of RAM. The 2D CNN and ViT models were evaluated on the ASL [15] and NUS-II [16] datasets.

## 5. Results

Each model, 2D-CNN and ViT, underwent 20 epoch training. It is essential to note that 20 iterations may not suffice to demonstrate ideal results, as observed in other studies. Still, they provide context crucial for assessing accuracy and effectiveness for future utilization.

Key metrics used to measure the quality of the trained model include recall, precision, and F1-score. In the context of gesture recognition, recall represents the proportion of individuals performing hand gestures in the test dataset that were correctly identified by the model and is determined by the formula:

$$Recall = \frac{TP}{TP + FN}, \tag{6}$$

where $TP$ represents the test result correctly indicating the presence of a condition or characteristic, and $FN$ represents the test result incorrectly indicating the absence of a certain condition or characteristic.

Precision signifies the proportion of individuals identified by the model as performing hand gestures who indeed perform hand gestures, defined by the formula:

$$Precision = \frac{TP}{TP + FP}, \tag{7}$$

where $TP$ represents the test result correctly indicating the presence of a condition or characteristic and $FP$ represents the test result incorrectly indicating the presence of a certain condition or characteristic.

The F1-score is calculated as the harmonic mean of precision and recall, as per formula (8). A higher F1 Score indicates better model performance. An ideal F1 Score of 1 signifies the model has perfect precision and recall.

$$F1 = 2 \cdot \frac{Precision \times Recall}{Precision + Recall}, \tag{8}$$

where precision and recall values are calculated using formulas (6) and (7).

Detailed results comparing subsets of gestures using 2D-CNN are presented in Table 1.

**Table 1**
**Experiment results for subsets of characters from the used datasets with the 2D-CNN architecture**

| | ASL dataset | | | NUS dataset | | |
|---|---|---|---|---|---|---|
| Symbol | Precision | Recall | F1 Score | Precision | Recall | F1 Score |
| A | 0.9078 | 0.9028 | 0.9053 | 0.9078 | 0.8634 | 0.8850 |
| B | 0.9189 | 0.9142 | 0.9165 | 0.9189 | 0.8742 | 0.8960 |
| E | 0.8757 | 0.6765 | 0.7633 | 0.8757 | 0.8526 | 0.8640 |
| G | 0.8643 | 0.6177 | 0.7205 | 0.8643 | 0.8485 | 0.8563 |
| I | 0.9052 | 0.9252 | 0.9151 | 0.9052 | 0.8532 | 0.8784 |
| P | 0.9154 | 0.9165 | 0.9160 | 0.9154 | 0.8521 | 0.8826 |
| S | 0.8843 | 0.8279 | 0.8552 | 0.8843 | 0.8378 | 0.8604 |
| Z | 0.8734 | 0.7644 | 0.8153 | 0.8734 | 0.8386 | 0.8557 |

The 2D-CNN demonstrated superior results for the ASL dataset, with an F1 Score averaging 0.881536 across all characters. This can be attributed to the isolated nature of the symbols on

which the model was trained, where the surrounding environment has less influence on the outcome. An average F1-score of 0.872366 was achieved on the NUS-II dataset.

Testing conditions for ViT were identical and yielded inferior results. For the ASL dataset, the F1 Score reached 0.848405, and for NUS-II, it was 0.843213. Interestingly, ViT exhibited better results in varied environmental conditions due to the self-attention mechanism inherent in transformers.

To improve ViT model performance, an additional 20 training epochs were conducted. Consequently, the average F1 Score across all characters increased to 0.880884.

Detailed results comparing subsets of gestures using ViT are presented in Table 2.

**Table 2**
**Experiment results for subsets of characters from the used datasets with the ViT architecture**

| | ASL dataset | | | NUS dataset | | | NUS dataset w/ 20 extra epochs | | |
|---|---|---|---|---|---|---|---|---|---|
| Symbol | Precis. | Recall | F1 | Precis. | Recall | F1 | Precis. | Recall | F1 |
| A | 0.9065 | 0.7989 | 0.8493 | 0.8765 | 0.7954 | 0.8340 | 0.9079 | 0.9028 | 0.9053 |
| B | 0.8838 | 0.8075 | 0.8439 | 0.9184 | 0.7835 | 0.8456 | 0.9189 | 0.9143 | 0.9166 |
| E | 0.9005 | 0.8073 | 0.8514 | 0.8852 | 0.7854 | 0.8323 | 0.8758 | 0.8166 | 0.8451 |
| G | 0.8994 | 0.8123 | 0.8536 | 0.9124 | 0.8048 | 0.8552 | 0.8644 | 0.7978 | 0.8297 |
| I | 0.8849 | 0.7978 | 0.8391 | 0.8905 | 0.7884 | 0.8364 | 0.9052 | 0.9253 | 0.9151 |
| P | 0.9059 | 0.8084 | 0.8544 | 0.9155 | 0.7979 | 0.8527 | 0.9155 | 0.9166 | 0.9160 |
| S | 0.8978 | 0.8043 | 0.8485 | 0.8773 | 0.7975 | 0.8355 | 0.8844 | 0.8279 | 0.8552 |
| Z | 0.8838 | 0.8131 | 0.8470 | 0.9054 | 0.8080 | 0.8539 | 0.8735 | 0.8545 | 0.8639 |

This revised section presents the results of the experiments, including metrics such as recall, precision, and F1-score, for ViT models. It also highlights the impact of additional training epochs on ViT model performance.

## 6. Discussion

The experimentation with the developed architectures for 2D-CNN and ViT in the context of gesture recognition has been conducted. Their performance based on the ASL and NUS-II datasets has been analyzed. The advantages and disadvantages of using these approaches for gesture recognition tasks have been highlighted. The superior effectiveness of 2D-CNN compared to ViT under limited resources has been demonstrated, with an F1-score of 0.881536 for 2D-CNN compared to 0.848405 for ViT, showcasing a 3.8% advantage for 2D-CNN.

The obtained results contribute to the further development of 2D Convolutional Neural Network (2D-CNN) and Visual Transformer (ViT) models for deep learning in hand gesture recognition. By showcasing the comparative performance of these architectures on gesture recognition tasks, this research adds to the body of knowledge on effective deep learning techniques for computer vision tasks, particularly in gesture recognition.

The practical significance of the obtained results lies in identifying advantages and drawbacks. The investigation of the effectiveness and efficiency of deep learning models 2D-CNN and ViT based on the ASL and NUS-II datasets, with the aim of further deployment in IT solutions for automated hand gesture recognition. By understanding the strengths and weaknesses of each model in the context of specific datasets, practitioners can make informed decisions about which architecture to choose for their particular application scenarios. Additionally, exploring these models' performance provides valuable insights for improving existing systems or developing new ones for gesture recognition applications in various fields, including human-computer interaction, healthcare, and augmented reality.

Moving forward, several avenues for future research can be considered based on the findings of this study. Firstly, further investigation into optimizing ViT models for gesture recognition

tasks could improve performance, potentially closing the performance gap with 2D-CNN architectures. Additionally, exploring larger and more diverse datasets could provide a more comprehensive understanding of these models' capabilities and limitations across different contexts and demographics. Furthermore, research into hybrid architectures that combine the strengths of both 2D-CNN and ViT could yield even better results, leveraging the spatial information captured by 2D-CNN with the self-attention mechanisms of ViT. Lastly, exploring real-time implementations and their performance in practical scenarios would be valuable for assessing the feasibility of deploying these models in real-world applications where low latency and high accuracy are essential.

# 7. Conclusions

When comparing convolutional neural network (CNN) and Vision Transformer models, significant differences in model size, memory requirements, accuracy, and productivity have been identified. CNN models are traditionally known for their compactness and efficient memory utilization, making them suitable for resource-constrained environments. They have demonstrated high effectiveness in image processing tasks and excellent accuracy in various computer vision domains. On the other hand, Vision Transformers offers a powerful approach for capturing global dependencies and contextual understanding in images, leading to improved performance in certain tasks. However, Vision Transformers typically have larger model sizes and higher memory requirements than CNNs. While they can achieve impressive accuracy, especially when working with large datasets, computational demands may limit their practicality in resource-constrained situations.

The choice between CNN and Vision Transformer models depends on specific task requirements, considering available resources, dataset size, and the trade-off between model complexity, accuracy, and productivity. As hand gesture recognition remains relevant, further research and refinement of both architectures will enable researchers and practitioners to make more informed decisions based on specific needs and constraints.

Continued investigation and improvement of both CNN and Vision Transformer architectures will empower researchers and practitioners to address evolving challenges in gesture recognition. Ultimately, this will advance state-of-the-art computer vision and enhance the practical applicability of these models in real-world scenarios.

# 8. References


[1] F. Zhan, Hand Gesture Recognition with Convolution Neural Networks, in: 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.

[2] P. Molchanov, S. Gupta, K. Kim & K. Pulli, Multi-sensor system for driver's hand-gesture recognition, in: 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), 1, 1-8. https://doi.org/10.1109/FG.2015.7163132

[3] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 223, 1725-1732. https://doi.org/10.1109/CVPR.2014.223

[4] O. Köpüklü, A. Gunduz, N. Kose and G. Rigoll, "Online Dynamic Hand Gesture Recognition Including Efficiency Analysis," IEEE Transactions on Biometrics, Behavior, and Identity Science, (2020), vol. 2, no. 2, 85-97, doi: 10.1109/TBIOM.2020.2968216.

[5] B. Su, P. Zhang, M. Sun et al., “Direction-guided two-stream convolutional neural networks for skeleton-based action recognition,” Soft Comput 27, 11833–11842 (2023). https://doi.org/10.1007/s00500-023-07862-1

[6] J. Yu, M. Qin & S. Zhou, “Dynamic gesture recognition based on 2D convolutional neural network and feature fusion,” Scientific Reports, 12(1), 1-15 (2022). https://doi.org/10.1038/s41598-022-08133-z
[7] N. Neverova, C. Wolf, G. W. Taylor, F. Nebout, Multi-scale Deep Learning for Gesture Detection and Localization. In: Agapito, L., Bronstein, M., Rother, C. (eds) Computer Vision - ECCV 2014 Workshops. ECCV 2014. Lecture Notes in Computer Science (2015), vol 8925. Springer, Cham. https://doi.org/10.1007/978-3-319-16178-5_33
[8] T. Yong, L. Kian, T. Connie, L. Chin-Poo, L. Cheng-Yaw, “Convolutional neural network with spatial pyramid pooling for hand gesture recognition,” Neural Computing and Applications, 33, 1-13 (2021). https://doi.org/10.1007/s00521-020-05337-0.
[9] T. Yong, L. Kian, L. Chin-Poo, “Hand Gesture Recognition via Enhanced Densely Connected Convolutional Neural Network,” Expert Systems with Applications, 175 (2021). https://10.1016/j.eswa.2021.114797.
[10] C. Osimani, J. J. Ojeda-Castelo, J. A. Piedra-Fernandez, “Point Cloud Deep Learning Solution for Hand Gesture Recognition,” International Journal of Interactive Multimedia and Artificial Intelligence (2023). https://doi.org/10.9781/ijimai.2023.01.001
[11] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, North American Chapter of the Association for Computational Linguistics (2019). https://doi.org/10.18653/v1%2FN19-1423
[12] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language Models are Unsupervised Multitask Learners, (2019) https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf
[13] Z. Hengshuang, J. Jiaya, K. Vladlen. (2020). Exploring Self-Attention for Image Recognition. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 10073-10082. https://doi.org/10.1109/CVPR42600.2020.01009.
[14] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko. (2020). End-to-End Object Detection with Transformers. https://10.1007/978-3-030-58452-8_13.
[15] A. L. C. Barczak, N. H. Reyes, M. Abastillas, A. Piccio, T. A. Susnjak. New 2D Static Hand Gesture Colour Image Dataset for ASL Gestures (2011).
[16] P. K. Pisharady, P. Vadakkepat, A. P. Loh, “Attention based detection and recognition of hand postures against complex backgrounds,” International Journal of Computer Vision (2013), 101, 403-419. https://doi.org/10.1007/s11263-012-0560-5
[17] Tan, Chun Keat, Kian Ming Lim, Roy Kwang Yang Chang, Chin Poo Lee, and Ali Alqahtani, "HGR-ViT: Hand Gesture Recognition with Vision Transformer" Sensors 23, no. 12: 5555 (2023). https://doi.org/10.3390/s23125555
[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit & N. Houlsby, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. (2020). ArXiv /abs/2010.11929
[19] V. Teslyuk, B. Borkivskyi, H. A. Alshawabkeh, Models and means of object recognition using artificial neural networks, in: MoMLeT+DS-2022: 4th International Workshop on Modern Machine Learning Technologies and Data Science, November 25-26, 2022, Leiden-Lviv, The Netherlands-Ukraine, CEUR Workshop Proceedings, 2022, 3312. https://ceur-ws.org/Vol-3312/paper20.pdf
[20] M. Nazarkevych, V. Hrytsyk, Y. Voznyi, A. Marchuk, and O. Vozna, Method of detecting special points on biometric images based on new filtering methods, in: Cybersecurity Providing in Information and Telecommunication Systems, 2021, Kyiv, Ukraine, CEUR Workshop Proceedings, 2021, 2923, 243–251.
[21] M. Nazarkevych, S. Dmytruk, V. Hrytsyk, et al., “Evaluation of the Effectiveness of Different Image Skeletonization Methods in Biometric Security Systems,” International Journal of Sensors, Wireless Communications and Control, 2021, 11(5), 542–552.
[22] N. Parmar, P. Ramachandran, A. Vaswani, I. Bello, A. Levskaya, J. Shlens, Stand-Alone Self-Attention in Vision Models, in: Advances in Neural Information Processing Systems 32, NeurIPS 2019: 68-80.