**Integrating YOLOv8 with Modified Squeeze-and-Excitation Network for Tomato Disease Detection and Fertilizer Recommendation**

Narra Ramesh Chandra1, Vedant Kishore Adka2, Manas Ranjan Prusty3

**1,2** School of Computer Science and Engineering, Vellore Institute of Technology, Chennai India.

3Centre for Cyber Physical Systems, Vellore Institute of Technology, Chennai, India.

# **Abstract**

Tomato plants are highly vulnerable to various diseases, which can severely impact their yield and overall quality. Traditionally, farmers rely on visual inspection to identify infections, but this method is often inaccurate, time-consuming, and labor-intensive. To overcome these challenges, this study explores a deep learning-based approach for tomato disease detection, utilizing an enhanced YOLOv8 model integrated with Convolutional Neural Networks (CNNs) for real-time identification. The model is further strengthened with Squeeze-and-Excitation (SE) and SE\* Blocks, which improve feature extraction and enhance classification accuracy. To ensure precise disease localization, the Segment Anything Model (SAM) is employed for leaf segmentation. Additionally, an HSV color space-based technique is incorporated to estimate disease severity, providing valuable insights for better crop management. A smart fertilizer recommendation system is also introduced, offering optimized treatment plans based on disease severity levels. By combining cutting-edge object detection, advanced segmentation, and automated disease management techniques, this approach aims to boost agricultural productivity and sustainability. Experimental results demonstrate a significant improvement in disease classification accuracy while maintaining efficiency, making the system well-suited for real-world farming applications.

**Keywords:** Tomato disease detection, deep learning, YOLOv8, CNN, Squeeze-and-Excitation Block, image segmentation, severity estimation, precision agriculture, SAM, real-time detection.

---------------------

\*corresponding author:

Manas Ranjan Prusty

manas.iter144@gmail.com

# **Introduction**

Tomato crops are very prone to a high number of diseases, which play a significant role in affecting their yield and quality. Successful disease intervention, control, and management rely on accurate and timely diagnosis. Currently, the disease is primarily visually examined by farmers or agriculture experts, which is a time-consuming, labor-intensive, and error-prone task. With the growing need for efficient and sustainable agriculture, computer vision-based deep learning methods have gained more popularity for the identification of plant diseases [3], [8]. These methods provide an efficient, scalable, and timely disease detection mechanism with minimized potential crop losses.

Deep learning models such as DenseNet and YOLO have been found to work exceptionally well on object detection and image classification [1], [11]. DenseNet has been applied extensively in medical and agricultural image processing due to its ability in robust feature extraction. Traditional deep learning models are, however, vulnerable to fine-grained disease classification, which leads to misclassification and false negatives [6]. To address this, classification models with Squeeze-and-Excitation (SE) Blocks have been suggested. The blocks enhance accuracy by enabling the network to focus on disease-related features.

To further enhance performance of classification, a compound Squeeze-and-Excitation Block (SE\*) with Channel-and-Spatial Attention mechanisms has been utilized. In contrast to the traditional SE Blocks that focus on channel-axis features, SE\* also captures spatial dependencies, enabling improved localization of disease areas. The two-level attention mechanism significantly improves classification performance, as evidenced in the new DenseNet-121 model, which outperforms other SE-augmented models [5]. For real-time disease detection, YOLOv8 has been utilized with SE and SE\* Blocks [11], [18]. YOLOv8 is also renowned for its speed in real-time object detection, ensuring rapid and accurate detection of disease with less computational overhead. Usage with SE\* to the YOLOv8 model enhances the model's capacity to concentrate on infected regions, enhancing classification efficiency without sacrificing inference speed. This renders the model highly beneficial for operational agricultural use where real-time detection is essential.

For effective localization of disease, YOLOv8 (Detection) is employed to identify individual leaves to precisely detect the disease. Instead of the conventional segmentation networks like U-Net, the Segment Anything Model (SAM) has been employed due to the improved segmentation accuracy. The zero-shot learning property of SAM allows accurate segmentation of complex disease patterns with limited large labeled sets [17]. This property improves the accuracy of classification even with changing environmental conditions, and SAM is thus highly appropriate under real-world agricultural environments.

In addition, an HSV color space-based system has also been incorporated for smart estimation of the severity of disease. This system categorizes the diseased area in terms of color differences and provides a measure of severity levels. Based on severity analysis, an intelligent fertilizer recommendation is proposed on the suggestion of optimal treatments to farmers for making informed decisions about crop management. With the inclusion of deep learning techniques for detection, classification, segmentation, severity estimation, and automated suggestions, the suggested system offers an overall solution for tomato disease monitoring [10]. Such an approach optimizes agricultural sustainability and productivity while enabling improved crop management practices.

The architecture of the proposed model can be seen in Figure 1.

The main contributions of this paper are as follows:

1. We propose a multimodal feature fusion network to combine complementary features from different modalities with skeleton features, which is largely unexplored for gesture recognition.
2. With the addition of a cross-attention layer, the improved MF-HAN model achieves competitive results in benchmark datasets with other state-of-the-art models, while maintaining a considerably lower computational complexity.

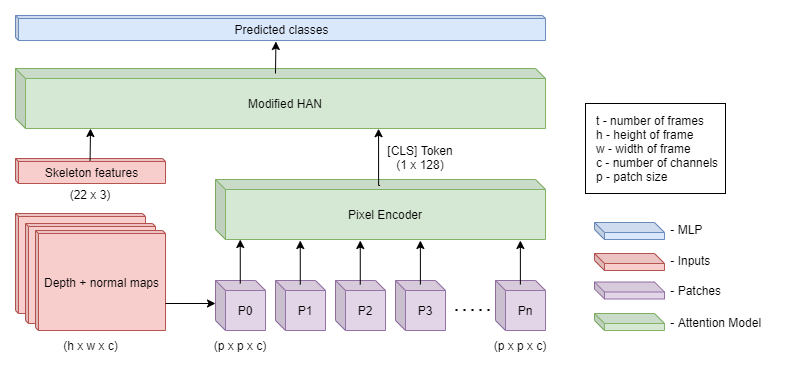


Figure 1 – An overview of the architecture of the proposed MFHAN models. The depth and normal maps are processed by a Pixel Encoder network and combined with the skeleton features in a modified HAN model to give out predictions.

# **Related work**

In literature review, various approaches have been attempted for detection and classification of tomato leaf disease, categorized according to feature extraction techniques, deep learning models, segmentation methods, real-time applicability, and explainable AI.

Several studies have made use of light-weight CNN models to enhance disease detection on low-end devices. Alnamoly et al. [1] introduced FL-ToLeD, an attention CNN model designed specifically for low-end devices, that has high classification accuracy and low computational overhead. Özbılge et al. [2] also introduced a Compact Convolutional Neural Network (CNN) designed particularly for tomato disease detection with efficiency as the primary concern without sacrificing accuracy.

Deep learning-based feature extraction techniques have also been widely researched. Hossain et al. [3] employed Deep Convolutional Neural Networks (DCNNs) for feature learning, demonstrating the effectiveness of CNN-based classifiers over traditional image-processing techniques. Hosny et al. [4] improved classification performance using the combination of Local Binary Patterns (LBP) and deep CNN features, demonstrating the utility of feature combination for multi-class disease classification.

Object detection algorithms like YOLO (You Only Look Once) have been applied widely for real-time disease detection. Umar et al. [5] enhanced YOLOv7 to detect multiple diseases in tomato plants with a significant boost in detection accuracy. Similarly, Liu et al. [11] developed an improved YOLOv8 model for detecting tomato leaf disease with enhanced object localization and detection speed compared to previous versions of YOLO. Brucal et al. [18] even optimized YOLOv8 with RoboFlow 2.0 to make it deployable in real-world agricultural applications.

For segmentation technique improvement, Ouamane et al. [6] proposed a pretrained CNN-based tensor subspace learning approach to improve segmentation accuracy for disease classification. Roy et al. [7] proposed a PCA DeepNet to further improve segmentation and have disease regions well separated. In contrast to the other contributions, Zhang et al. [8] also addressed object detection improvement with the introduction of deep learning-based bounding box regression for better segmentation accuracy.

Real-world applications require efficient and reliable mobile-based models. Ndovie and Masabo [9] used MobileNetV3 for in-field detection of tomato disease with high accuracy and reduced computational cost, hence it can be implemented on mobile phones. De Silva and Brown [10] employed multispectral imaging using deep learning models and introduced the benefit of multi-channel image inputs to classify diseases well.

Early detection of diseases is necessary for effective crop management. Mamatha and Raju [12] pointed out preprocessing techniques to enhance early disease detection through image enhancement and noise reduction. Halder et al. [13] utilized deep neural networks for the early detection of tomato leaf blight disease symptoms, which showed that early detection significantly reduces crop loss.

Hybrid deep learning model strategies have also been explored. Verma et al. [14] proposed a Hybrid LSTM-CNN model that utilizes spatial and temporal features to perform multi-class classification. Syed et al. [15] tackled overlapping diseases by employing CNN-based feature extraction, which enables classification to be properly performed even when there are overlapping disease. Khan et al. [16] proposed TomFormer, a transformer-based approach for early and precise detection of tomato leaf disease.

The year also witnessed efforts on disease severity estimation and explainable AI. Yashu et al. [17] suggested DeepHarvest, a hybrid model to predict disease severity levels, generating actionable knowledge in precision agriculture. Gowri et al. [19] undertook explainable AI-based model interpretability, making the deep learning model decision-making process clear and trustworthy to agricultural experts.

Besides these advancements, another prominent paper by [20] proposed an early image-based classification model using Deep Neural Networks (DNNs). The paper highlighted how early disease detection from high-quality image data and well-optimized neural network structures can actually enhance crop health monitoring. In general, improvements in CNN-based models, object detection models (YOLO), segmentation methods, hybrid deep learning models, and explainable AI have collectively helped in better detection and classification of tomato leaf diseases. Nevertheless, there remain some challenges in managing overlapping diseases, enhancing real-time performance, and enhancing model interpretability. Our work is based on these improvements by utilizing cutting-edge deep learning methods for boosting accuracy, efficiency, and real-world applicability of tomato disease detection models.

# **Proposed Methodology**

The overall architecture of the proposed model is shown in Figure 1**.** Given a hand gesture represented by skeleton features and the depth map, normals maps are generated from the depth maps and the combination of depth and normal maps are passed into the *Pixel Encoder* to generate embeddings that represent the action. The skeleton features along with these embeddings are used in the HAN model to classify the dynamic gesture. The embeddings are added to the baseline HAN model, which is designed to work with only skeleton features, to also incorporate image features to create a rich representation of the gesture.

## *3.1. Pixel Encoder Network*

The Pixel Encodernetwork works on the combination of depth and normal maps of a gesture , by performing patch-level operations on the images. The architecture of the Pixel Encoder network can be seen in Figure 2. The feature maps are split into patches of shape each and are tokenized using apixel tokenization algorithm from [9]. It is used to identify discriminative features in the hand images which is very important for fine gestures. Since, multiple modalities can be used as input, a convolution layer with a kernel size of 3, stride and padding of 1 is used to reduce the number of channels in the input features to 1. The tokenization is controlled by two learnable parameters namely WA and WB. The following set of equations from [9] summarizes the process:

|  |  |
| --- | --- |
|  | (1) |
|  |  |
|  | (2) |
|  |  |
|  | (3) |
|  |  |
|  | (4) |
|  |  |
|  | (5) |

Here, represents the ith patch extracted from the image features , , , , , and . The pixel tokens of each patch are concatenated into a list of tokens .

The encoder is the same as the encoder block in [22], which uses a self-attention mechanism as shown in (a) of Figure 2. , which is the concatenation of the pixel tokens along with a learnable class token , is added to the sinusoidal positional embeddings of the encoder and subjected to the encoder of the transformer architecture. The class token after the input is processed is the embedding vector describing the image features of an entire gesture sequence.

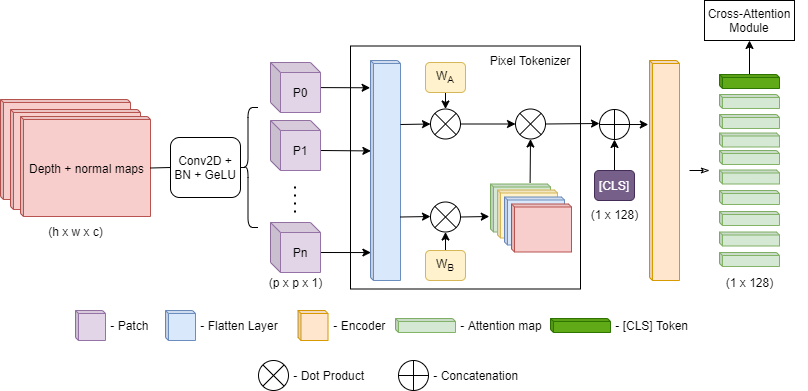


Figure 2 - Architecture of the Pixel Encoder network.

## *3.2. Modified Hierarchical Self-Attention Network (HAN)*

The Hierarchical Self-Attention Network (HAN) model from [8] is modified as a dynamic hand gesture recognition model to use multimodal features instead of only skeleton features. The structure of HAN is designed in accordance to the hierarchical nature of the anatomy of a hand, increasing in complexity from joints to fingers and to the entire palm. Each of these anatomical parts has a corresponding attention layer in the model that focuses on the skeleton features of that part. Joint attention module focuses on the joints of each finger, while the Finger attention module focuses on the complete set of fingers. The Temporal attention module focuses on the variation along time for the entire hand, while the Fusion attention module acts on the temporal relation information from the previous layer to generate predictions. Each attention layer is multi-headed and uses self-attention, which improves the training stability of the model. In self-attention, , where Q stands for Query, K stands for Key and V stands for Value. For HAN, this means that the same input to the attention layer is used to compute Q, K and V. This makes learning representations and relations from parts of the data to itself, and in this case, evolution of hand features across frames. The layers also share weights across the different fingers and frames of the data sequence. The joint attention module shares weights across all the joints, finger attention module across all the fingers and temporal attention module across all the frames. The model also has a two-stream variant that uses HPEV, FRPV and HMM features [19] derived from the skeleton features alongside the outputs of the fusion network using a late fusion strategy.

The model, however, only processes skeleton features, which makes the fusion of information from other modalities highly viable. The model presents two places for data fusion – one in the fusion attention module (late fusion), which is also used in the two-stream variant of the model and also in the finger attention layer, which operates on the entire hand. The latter was used in the proposed method due to only hand features being available from the images, while the former requires data processed on the temporal scale.

## *Finger cross-attention module*

Cross-attention was introduced in [22] in the decoder architecture but was not named so until recently. It involved the usage of different feature vectors in the generation of queries, keys and values in the attention module. This facilitated the incorporation of information from a completely different modality, in this case image features, in addition to the skeleton features, resulting in the generation of richer representations for each gesture. Figure 2 shows the difference between a self-attention network and a cross-attention network.

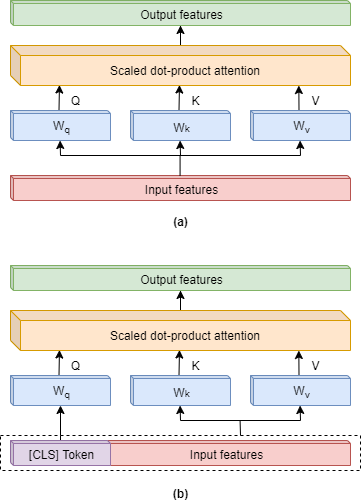


Figure 3 - Visualization of the different types of attention networks being used in the proposed approach. (a) Self-Attention network (b) Cross-Attention network.

In our proposed methodology, we used a multiheaded cross-attention module with residual connections in place of the finger attention module in HAN [8], to incorporate image features into the feature representations. The idea was to use the image features as a complementary set of features to augment the skeleton features, taking inspiration from the patch-based cross-attention in [9]. For this, the [CLS] token from the *Pixel Encoder* - is used in generating the queries, while the skeleton features are used in generating the keys and values. Figure 3shows the finger cross-attention block in the proposed methodology and how it differs from the finger-attention block used in the HAN model. This was especially useful in the two-stream version of the model, where the fusion network is already employed in the late fusion of multiple feature representations [19].The following equations show the mechanism of computing values for cross-attention:

|  |  |
| --- | --- |
|  | (6) |

Here, represents the learnable weights of the multiheaded cross-attention module.

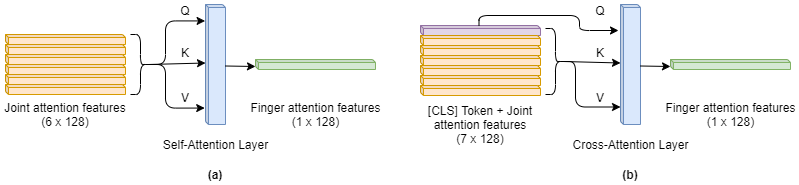


Figure 4 - Visualization of the finger attention block. (a) Finger attention block in the HAN model which applies self-attention on joint features. (b) Joint attention in the proposed approach which applies cross-attention on the joint features and the depth features.

# **Experiment Details**

## *Dataset*

SHREC’17 Track dataset provides depth images as well as skeleton joints data collected using a short-range Intel RealSense depth sensor. The dataset contains 2800 sequences of 14 hand gestures performed by 28 individuals in two ways – using a single finger and the whole hand, as seen in Figure 3. The gestures in the dataset are divided into coarse (characterized by hand motion) and fine (characterized by hand shape). The training set of the dataset has 1960 sequences and the test set has 840 sequences respectively. Each sequence has the coordinates of the 22 joints in the 2D depth image space and in the 3D world space and each frame of the sequence frame is a depth map of resolution 640x480 pixels.

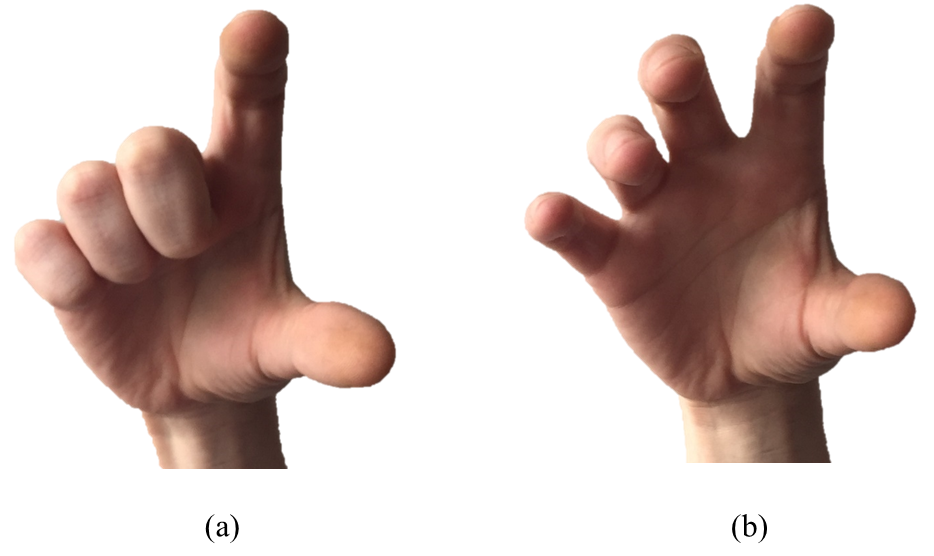


Figure 5 - Gesture types in SHREC'17 track dataset. (a) Gesture performed using a single finger. (b) Gesture performed using the whole hand.

Table 1 - List of hand gestures and their types in the SHREC'17 track dataset.

|  |  |  |
| --- | --- | --- |
| **Name of the gesture** | **Gesture code** | **Type of the gesture** |
| Grab | G | Fine |
| Tap | T | Coarse |
| Expand | E | Fine |
| Pinch | P | Fine |
| Rotation Clockwise | RC | Fine |
| Rotation Counter Clockwise | RCC | Fine |
| Swipe Right | SR | Coarse |
| Swipe Left | SL | Coarse |
| Swipe Up | SU | Coarse |
| Swipe Down | SD | Coarse |
| Swipe X | SX | Coarse |
| Swipe + | S+ | Coarse |
| Swipe V | SV | Coarse |
| Shake | S | Coarse |

## *Data Processing*

Each frame of the video sequence for a gesture contains a depth map of resolution 640x480 pixels, which is denoted by . Surface normals map can be calculated from D by finding the normalized gradients across horizontal and vertical directions. The gradients and are calculated by convolving the Sobel’s kernels on the depth map D, as shown below:

The surface normal vectors map is considered to be . The surface normals map is then calculated by normalizing. The resultant processed depth and normal maps can be seen in Figure 1. Each depth map also comes with a bounding box that describes the ROI of the hand gesture for that particular frame. The bounding box can either be used as a mask to retain information only within the ROI or can be used to crop out the ROI, applied on both the depth map D and the surface normals map . This produces two variations of the dataset – un*cropped* and *cropped* respectively, which can be compared for effective feature extraction feasibility. Furthermore, *grayscale normalization* is applied on the depth map, following [21] by splitting the gray levels into n discrete bands to increase contrast, as the low contrast of depth maps do not provide scope for distinct features. The cropped images are resized to a resolution of 50x50 while the uncropped images are resized to a resolution of 128x96.

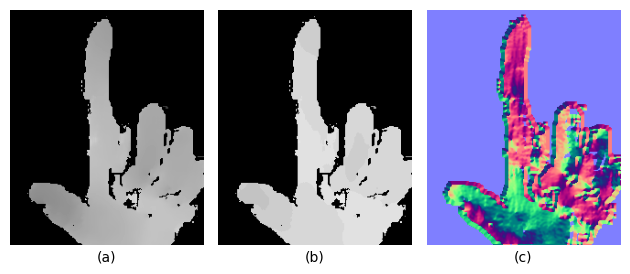


Figure 6 - Cropped hand gesture frames from SHREC'17 track dataset. (a) Un-processed depth map (b) Processed depth map by applying Grayscale Variation (c) Extracted normal map by applying Sobel's filter on the depth maps.

## *Data Augmentation*

For the purposes for comparing between the backbone model, the augmentations are almost the similar as the ones used in [8]. Random shifts in the range of 20% of the image dimensions, random scaling in the range of 20% of the image dimensions and time interpolation between the frames of the gesture with a random smoothing factor between 0 and 1 are used. It is also decided to maintain a 50% chance of applying these augmentations on the patches, to make it easier for the model to learn about the actual gesture motions and the temporal aspects attached to it. For ease of implementation and execution, the patches containing the gestures are extracted and saved, while the augmentations are performed during training.

## *Implementation Details*

All experiments are conducted using PyTorch on a Nvidia Tesla P100 GPU. Each experiment was run for 3 times and the best metrics are reported. For the underlying HAN model, the recommended hyperparameters from [8] were used. The same hyperparameters were also used for the attention block in the *Pixel Encoder* network - 8 attention heads with a feature dimension of 32 and a dropout probability of 0.1. The patch size is 10x10 for cropped images and 16x16 for uncropped images. The feature dimension for the patch-attention network is assumed to be 128 to match the feature dimensions of the HAN model. 8 frames were sampled per gesture using a sampling technique from [17], which splits the sequences into 8 equal parts and selects one frame from each split randomly as can be seen from Figure 2, where 8 frames are sampled from a sequence containing 63 frames. If the number of the frames in the sequence is less than 8, then the sequence is padded with empty frames. Data augmentation performed on the images include shifting, scaling and time interpolation. The optimizer is chosen to be AdamW with an initial learning rate of 0.001 and a weight decay of 0.1. The loss function is chosen to be cross-entropy with label smoothing of 0.1. The label smoothing acts as regularization and helps to prevent over-fitting of the model. Batch size is set to 32 for both training and testing. Linear warmup is applied on the learning rate for the first 10 epochs and reduces the learning rate by a factor of 10 when learning stagnates for 50 epochs. The training ends when the learning rate drops 4 times.

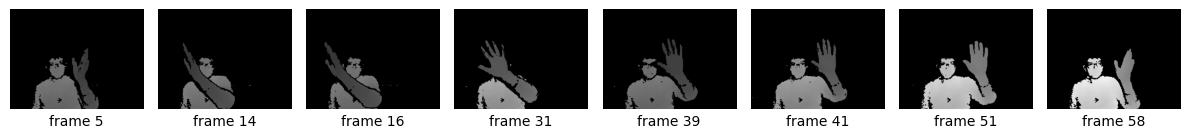


Figure 7 – Sampling technique applied on a Slide-Right sequence containing 63 frames in the SHREC’17 track dataset. 8 frames are sampled from the sequence by splitting the frames into 8 windows and choosing a random frame from each window.

## *Ablation Study*

We also examined the effect of cropping the depth maps to the hand gesture ROI by using cropped and uncropped versions of the depth maps and corresponding normal maps. This was done to explore the effect of uncropped images in providing motion paths for global attention, which cropped images would lack. We also tried removing the cross-attention block and its residuals separately to understand and quantify the impact of the cross-attention layer in the proposed model. In this setting, the encoded feature representation from the *Pixel Encoder* would still be appended to the joint attention features, although the features would only undergo self-attention and not cross-attention.

# **Discussions**

## *Comparison with state-of-the-art models*

For fairness in comparison, the single stream version of our proposed approach will be compared only with other single stream approaches, while the two-stream version will be compared against its counterparts. Our MFHAN outperforms all traditional approaches like De Smedt et. al. [1] with a very large margin in accuracy. The classification accuracy on both 14 and 28 gesture settings is higher than other convolution and recurrent models like STA-Res-TCN and its baseline Res-TCN [7], gVar-FL-Fusion [21] and MFA-Net [23], which we theorize is due to the patch-level attention provided by the vision transformer inspired *Pixel Encoder*. It also out-performs graph neural networks with attention mechanisms like HG-GCN **[cite]** and achieves almost comparable results with DG-STA [4] in 14 gestures setting, but out-performs it by almost 1.56 % in the 28 gestures setting. Furthermore, while the accuracy under 14 gesture setting may have deteriorated from the baseline HAN [8] model by a whole 1%, the 28 gesture setting witnessed a rise in accuracy by about 1.19%, which shows that the model has learned to identify and learn the intra-gesture variance across two variations of a same gesture.

Table 2 - Comparison of MF-HAN with single-stream models on SHREC’17 track dataset with 14 and 28 gesture settings.

|  |  |  |
| --- | --- | --- |
| **Approach** | **Accuracy (in %)** | |
| **14 G** | **28 G** |
| De Smedt Et.al. [1] | 88.24 | 81.9 |
| Res-TCN [7] | 91.1 | 87.3 |
| MFA-Net [23] | 91.31 | 86.55 |
| HG-GCN [24] | 92.8 | 88.3 |
| gVar-FL-Fusion [21] | 93.33 | 90.24 |
| STA-Res-TCN [7] | 93.6 | 90.7 |
| DG-STA [4] | 94.4 | 90.7 |
| HAN [8] | **95** | 91.07 |
| *MFHAN (Proposed Model)* | 93.93 | **92.26** |

Our two-stream version of the proposed model MFHAN-2S showed more than promising results, compared to the single-stream version MFHAN. The accuracy in the 14 gestures setting is almost comparable to [19], which uses 2 streams of hand-crafted hand movement and posture variation features, while HAN-2S [8] and Multi Input fusion lightweight network [25] show superior results. But the proposed model achieves the best accuracy in the 28 gestures setting, surpassing the best model [8] by more than 0.4%.

Table 3 - Comparison of MFHAN-2S with two-stream models on SHREC'17 dataset with 14 and 28 gesture settings.

|  |  |  |
| --- | --- | --- |
| **Approach** | **Accuracy (in %)** | |
| **14 G** | **28 G** |
| HPEV+HMM+FRPV [19] | 94.88 | 92.26 |
| HAN-2S [8] | 95 | 92.86 |
| Hu et. al. [25] | **95.80** | 92.5 |
| *MFHAN-2S (Proposed Model)* | 94.17 | **93.21** |

## *Ablation Study*

From Table 4we can see the effect of different components in the cross-attention module of the proposed model. A significant drop in accuracy can be seen when the cross-attention module is substituted with a base attention module. Especially, the accuracy in 28 gestures task drops by 2%. The effect of removing residual connections in the cross-attention module has minimal effect in the 14 gestures task, which we believe that the joint attention module compensates for. However, there is a significant drop without residual connections in 28 gestures task, which indicates that it is a vital contributor to the increased recognition when the number of gestures are increased. Moreover, from Figure 6, the model trained with cross-attention converges in about 233 epochs, while the models with no residuals and cross-attention take 294 and 305 epochs respectively, solidifying the claim that cross-attention helps in faster convergence of the model, while also providing multimodal discriminating information for the model to learn better.

Table 4: Ablation study of cross-attention module and residuals in MF-HAN on SHREC’17 track dataset 14 and 28 gestures settings.

|  |  |  |
| --- | --- | --- |
| **Ablation group** | **Accuracy (in %)** | |
| **14 G** | **28 G** |
| MF-HAN | **93.93** | **92.26** |
| MF-HAN without residuals | 93.45 | 90.12 |
| MF-HAN without cross-attention | 92.86 | 90.71 |

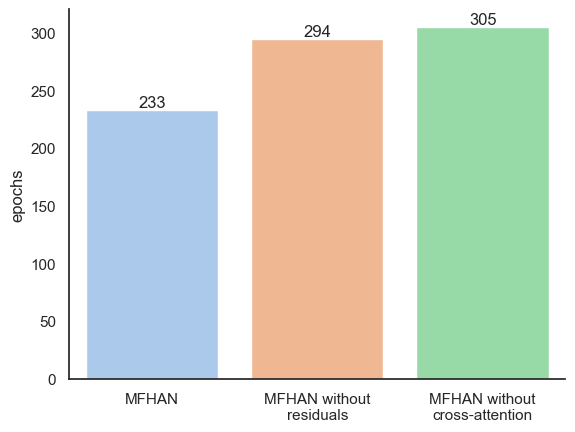


Figure 8 - The number of epochs for training of the models considered in the cross-attention ablation study.

Table 5shows the effect of using cropped and uncropped frames in the *Pixel Encoder* network of the MF-HAN model. Cropped images show almost the same performance in the 14 gestures setting but show a significant improvement in the 28 gestures setting when compared to the uncropped images. We theorize that the sparse nature of the uncropped images was not able to provide the necessary features for the attention blocks to capture. The cropped images provided a higher resolution of features with less sparsity, which is very useful in learning discriminating features for an increased set of classes.

Table 5: Ablation study of using cropped and uncropped frames for feature processing in MF-HAN on SHREC'17 track dataset 14 and 28 gestures settings.

|  |  |  |
| --- | --- | --- |
| **Ablation group** | **Accuracy (in %)** | |
| **14 G** | **28 G** |
| Cropped | **93.93** | **92.26** |
| Uncropped | 92.86 | 89.64 |

## *Improvements & Results*

Table 6shows a performance comparison of MFHAN and MFHAN-2S with the baseline HAN model on the coarse and fine gesture categories of SHREC’17 track dataset. The baseline model shows the highest accuracy in the coarse gestures in 14 gestures setting, while our proposed MFHAN-2S outperforms both HAN as well as our MFHAN model in both fine and coarse gesture categories in the 28 gestures setting.

Table 6 - Comparison of gesture classification accuracy for fine and coarse categories between the HAN model and the proposed models on 14 and 28 gesture settings of the SHREC'17 track dataset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Approach** | **14 Gestures Accuracy (in %)** | | **28 Gestures Accuracy (in %)** | |
| **Fine** | **Coarse** | **Fine** | **Coarse** |
| *HAN* | 92.78 | **96.09** | 83.01 | 91.31 |
| *MFHAN* | **93.14** | 94.32 | 88.45 | 94.14 |
| *MFHAN-2S* | 91.70 | 95.74 | **89.17** | **95.91** |

Table 7 - Performance metrics of the proposed models on the 14 and 28 gestures settings of the SHREC'17 track dataset.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Approach** | **Accuracy** | **Precision** | **Recall** | **F1** | **AUC** |
| **(a)** **14 gestures setting** | | | | | |
| *MFHAN* | 0.9393 | 0.9414 | 0.9386 | 0.9386 | 0.9951 |
| *MFHAN-2S* | 0.9417 | 0.9449 | 0.9403 | 0.9413 | 0.9939 |
| **(b) 28 gestures setting** | | | | | |
| *MFHAN* | 0.9226 | 0.9246 | 0.9211 | 0.9211 | 0.9945 |
| *MFHAN-2S* | 0.9321 | 0.9331 | 0.9303 | 0.9304 | 0.9961 |

Table 7shows the best metrics for 14 and 28 gestures settings of our MFHAN and MFHAN-2S models, both showing a very AUC score and sufficiently high scores across the board. In all the fields, our MFHAN-2S outperforms MFHAN in both the gesture settings. Figure 7 shows a confusion matrix for the predictions under 14 gestures setting of MFHAN-2S. It can be observed that our model is able to correctly distinguish between the different classes as is evident from the main diagonal. The false positives are also comparatively low: for example, the model sometimes misclassifies the *Grab* gesture as a *Tap* or *Pinch* gesture, etc. The false negatives are also under controlled thresholds. The same trend is seen in Figure 8, which shows the confusion matrix for the same model under 28 gestures classification. It can also be seen that the model sometimes confuses *Rotation (Clockwise)* with *Rotation (Counter-Clockwise)*, which could mean that the model requires more directional information in the temporal features. It is also interesting to note that the model follows the same pattern of misclassifying *Grab* and *Pinch* gesture, as the 14-gesture setting model, indicating that more focus on discriminating features for these two specific gestures are required.

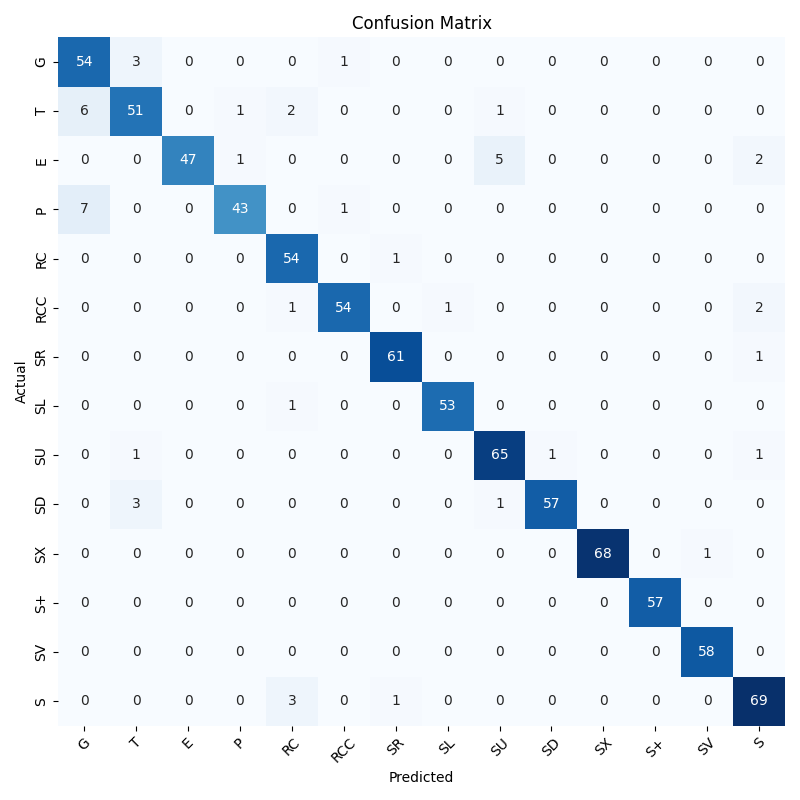


Figure 9 - Confusion matrix for predictions of MFHAN-2S on the 14 gestures setting of the SHREC'17 track dataset.

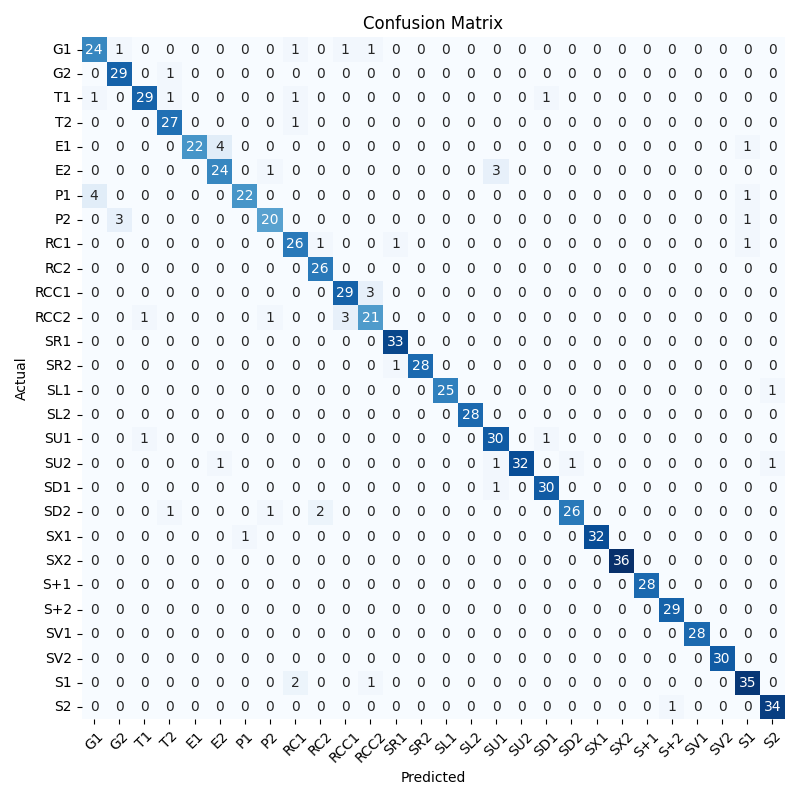


Figure 10 - Confusion matrix for predictions of MFHAN-2S on the 28 gestures setting of the SHREC'17 track dataset.

# **Conclusion**

This paper proposed a multimodal fusion model to incorporate depth features and skeleton features for the task of dynamic hand gesture recognition. Our method modifies the HAN [8] model use cross-attention in the finger attention module, which receives a context vector from a transformer encoder based *Pixel Encoder* model inspired from [9]. The *Pixel Encoder* model captures spatial information from the depth and normal maps of each frame in the sequence, which is used in cross-attention as a complementary source of information to the skeleton features. While the approach shows deterioration in classification accuracy for the 14 gestures setting, it outperforms other state-of-the-art models in the 28 gestures setting on the SHREC’17 track dataset. Moreover, the two-stream version of the model shows the same trend in performance with superior results as compared to other one and two stream models in the 28 gestures setting. From the above-mentioned results in the 28 gestures setting, it is evident that the approach works well in boosting the model’s capability to quantify intra-class and inter-class differences in an increased class set. In the proposed approach, an untrained encoder network is used in the *Pixel Encoder* model to be trained from scratch along with the rest of the network. In the future, transfer learning can be explored by using pretrained Vision Transformers (ViT) or ResNet models.

# **Acknowledgement**

I would like to sincerely acknowledge Vellore Institute of Technology, Chennai, India for their amazing support and infrastructure during the course of this work. I would also like to thank the School of Computer Science and Engineering for their support.

# **References**

[1] Q. De Smedt, H. Wannous, J.-P. Vandeborre, J. Guerry, B. Le Saux, and D. Filliat, “Shrec’17 track: 3d hand gesture recognition using a depth and skeletal dataset,” in *3DOR-10th Eurographics Workshop on 3D Object Retrieval*, 2017, pp. 1–6.

[2] M. Ur Rehman *et al.*, “Dynamic hand gesture recognition using 3d-cnn and lstm networks,” *Comput. Mater. Contin.*, vol. 70, no. 3, 2021.

[3] N. L. Hakim, T. K. Shih, S. P. Kasthuri Arachchi, W. Aditya, Y.-C. Chen, and C.-Y. Lin, “Dynamic hand gesture recognition using 3DCNN and LSTM with FSM context-aware model,” *Sensors*, vol. 19, no. 24, p. 5429, 2019.

[4] Y. Chen, L. Zhao, X. Peng, J. Yuan, and D. N. Metaxas, “Construct dynamic graphs for hand gesture recognition via spatial-temporal attention,” *ArXiv Prepr. ArXiv190708871*, 2019.

[5] K. Yang, R. Li, P. Qiao, Q. Wang, D. Li, and Y. Dou, “Temporal pyramid relation network for video-based gesture recognition,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 3104–3108.

[6] A. D’Eusanio, A. Simoni, S. Pini, G. Borghi, R. Vezzani, and R. Cucchiara, “A transformer-based network for dynamic hand gesture recognition,” in *2020 International Conference on 3D Vision (3DV)*, 2020, pp. 623–632.

[7] J. Hou, G. Wang, X. Chen, J.-H. Xue, R. Zhu, and H. Yang, “Spatial-temporal attention res-TCN for skeleton-based dynamic hand gesture recognition,” in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018, pp. 0–0.

[8] J. Liu, Y. Wang, S. Xiang, and C. Pan, “Han: An efficient hierarchical self-attention network for skeleton-based gesture recognition,” *ArXiv Prepr. ArXiv210613391*, 2021.

[9] S. K. Roy, A. Deria, D. Hong, B. Rasti, A. Plaza, and J. Chanussot, “Multimodal fusion transformer for remote sensing image classification,” *ArXiv Prepr. ArXiv220316952*, 2022.

[10] M. Munasinghe, “Dynamic hand gesture recognition using computer vision and neural networks,” in *2018 3rd International Conference for Convergence in Technology (I2CT)*, 2018, pp. 1–5.

[11] H.-Y. Chung, Y.-L. Chung, and W.-F. Tsai, “An efficient hand gesture recognition system based on deep CNN,” in *2019 IEEE International Conference on Industrial Technology (ICIT)*, 2019, pp. 853–858.

[12] H. Tang, H. Liu, W. Xiao, and N. Sebe, “Fast and robust dynamic hand gesture recognition via key frames extraction and feature fusion,” *Neurocomputing*, vol. 331, pp. 424–433, 2019.

[13] R. Jain, R. K. Karsh, and A. A. Barbhuiya, “Encoded motion image-based dynamic hand gesture recognition,” *Vis. Comput.*, vol. 38, no. 6, pp. 1957–1974, 2022.

[14] B. Verma and A. Choudhary, “Dynamic Hand Gesture Recognition using Convolutional Neural Network with RGB-D Fusion,” in *Proceedings of the 11th Indian Conference on Computer Vision, Graphics and Image Processing*, 2018, pp. 1–8.

[15] M. Abavisani, H. R. V. Joze, and V. M. Patel, “Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1165–1174.

[16] Y. Ma, T. Xu, and K. Kim, “Two-Stream Mixed Convolutional Neural Network for American Sign Language Recognition,” *Sensors*, vol. 22, no. 16, p. 5959, 2022.

[17] Z. Cao, Y. Li, and B.-S. Shin, “Content-Adaptive and Attention-Based Network for Hand Gesture Recognition,” *Appl. Sci.*, vol. 12, no. 4, p. 2041, 2022.

[18] Q. De Smedt, H. Wannous, and J.-P. Vandeborre, “Skeleton-based dynamic hand gesture recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 1–9.

[19] J. Liu, Y. Liu, Y. Wang, V. Prinet, S. Xiang, and C. Pan, “Decoupled representation learning for skeleton-based gesture recognition,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5751–5760.

[20] P. N. Huu and T. L. Ngoc, “Two-stream convolutional network for dynamic hand gesture recognition using convolutional long short-term memory networks,” *Vietnam J. Sci. Technol.*, vol. 58, no. 4, pp. 514–523, 2020.

[21] H. Mahmud, M. M. Morshed, and M. Hasan, “A deep-learning–based multimodal depth-aware dynamic hand gesture recognition system,” *ArXiv Prepr. ArXiv210702543*, 2021.

[22] A. Vaswani *et al.*, “Attention Is All You Need.” arXiv, Dec. 05, 2017. Accessed: Jan. 13, 2023. [Online]. Available: http://arxiv.org/abs/1706.03762

[23] X. Chen, G. Wang, H. Guo, C. Zhang, H. Wang, and L. Zhang, “Mfa-net: Motion feature augmented network for dynamic hand gesture recognition from skeletal data,” *Sensors*, vol. 19, no. 2, p. 239, 2019.

[24] Y. Li, Z. He, X. Ye, Z. He, and K. Han, “Spatial temporal graph convolutional networks for skeleton-based dynamic hand gesture recognition,” *EURASIP J. Image Video Process.*, vol. 2019, no. 1, pp. 1–7, 2019.

[25] Q. Hu, Q. Gao, H. Gao, and Z. Ju, “Skeleton-Based Hand Gesture Recognition by Using Multi-input Fusion Lightweight Network,” in *International Conference on Intelligent Robotics and Applications*, 2022, pp. 24–34.