CONVOLUTION ADDITIVE APPROACH TOWARDS SIGN LANGUAGE RECOGNITION USING VISION TRANSFORMERS

Bimsara Pasindu Gallage (HNDDS23.2 F-012)

BSc(Hons) Data Science

National Institute of Business Management(NIBM) - NIC

Colombo, Sri Lanka

A project report submitted to the national Institute of Business management for the partial fulfillment of requirement of the Higher National Diploma in Data Science.

DECLARATION

I hereby declare that the work presented in this project report was caried out independently by myself and have cited the work of others and given due reference diligently.

……

Bimsara Pasindu Gallage

……..

Date

I certify that the above student carried out his/her project under my supervision and guidance.

……

Supervisor

(Mr. Ashan Arthanayake/

Mrs. Chamilanka Wanigasekara)

Acknowledgement

I would like to give my gratitude to Mr. Ashan Arthanayake for taking time to review my project ideas and guiding and correcting my work throughout the time and being my mentor towards come up with quality works always. Also, I would like to thank our dean Dr. Wijayasiri for giving us valuable lectures on how to carry on research, the meaning behind knowledge discoveries and proper handling on project structure step by step. I would like to thank my family members for always being there for my academic success. Finally, I am grateful for all the relations that guided me to better academic life by staying besides me.

Abstract

Hand gesture recognition systems has been a topic that were rice in the past few years due to his its numerous applications and value towards the tools for disabled people and much more applications. As for the deep learning approach towards building successful hand sign recognition models convolutional neural network bracket CNN has done a great contribution towards it building various architectures to extract the different static and dynamic hand signs from given images. Even though CNN models have been able to get accurate performance using their architectures vision Transformers came into the field with their advance attention over image. Vision Transformers have their long-range capturing capabilities since it uses partitioning images into patches, vectorize them and calculate self-attention between patches. Even though their mechanism can upgrade the classification task using increasement of the vector embeddings and are there parameter tuning its computational complexity and capacity requirement is increasing quadratic way due to the complex mechanism of dot product similarity score calculation. A latest research paper has introduced much simpler way to calculate the similarity scores and, in this paper, mechanism called ‘Convolution additive self-attention” has implemented parallel to the traditional first vision transformer model and evaluated they are efficiencies and accuracies for American Sign Language data set. Even though both modules performed very high accuracy in all three training, test and validation phases in terms of efficiency novel approach of convolutional additive attention mechanism had overpowered the traditional vision transform approach with marking half the time for the total runtime of the model and half time for predicting the class label of a sample compared to the traditional model.

Table of contents

1. Introduction

1.1 Background

Hand gesture recognition is a topic that has been very popular within last few years since that is one of major methodology to communicate between humans. Those various types of hand gestures which are used in number of situations to interpret various meanings has got it a special place in general communication. There are numerous amounts of sign languages which are called lexicons that has different origins, different use cases and different interpretation based on the signs they are showing. As a sign language which has a huge usage compared to the population density, the American Sign Language lexicon got a considerable attention towards it throughout the years. Since the main task involves capture various movements from a given image or image sequence when it comes to hand gesture recognitions, convolutional neural network architectures have had a major contribution towards building models for those recognition task successfully. The biggest challenge for this image classification task is when they are growing into larger number of classes the more and more layers, or the parameters must add to the model in order to capture they are class or meanings accurately. Although they were some modifications to the CNN based recognition models, they were always limited to the localized pattern which they were extracting from the original images. Therefore, the long-range dependencies of vision Transformers which were formed very lately with its architecture of token mixture method that helps to communicate between each part of the original image had a very good chance in all the classification problems even though they are record a higher computational capacity compared to traditional CNN methods. Number of modifications on vision Transformers has been implemented and successful towards reducing computational requirement of building it’s kind of models.

1.2 Research Problem

The hand gesture recognition models have gradually increased over the time based on CNN architectures to recognize hand has obtained comparably a higher accuracy but as the main limitation of using weights-biases based pattern recognizing CNN architectures is that they are adjusted on limited weights and biases to adjust them according to various capturing patterns and yet with the various conditions. These models became larger over years due to reason of maintain the accuracy consistently throughout changes conditions. This generalization issue is overcome by the vision transformers (ViTs) but since it demands more computational power in mathematical operations which happens simultaneously it's far behind the efficient score in research for tasks compared to CNN architectures. Also, the path of hand gesture recognition tasks approached by division Transformers separately end similarity calculation efficient Transformers build separately the combination of these two haven't implemented together for this sign language recognition. Implementation of it and evaluation can lead to number of results based on their outputs.

1.3 Objectives of the Project

1. Building A Convolution Additive Vision Transformer Base Model for American Hand Sign Recognition
2. Evaluating Model Efficiency with Flop Count and Other Measurements Using Traditional multi-head self-attention Vision Transformer

1.4 Research Questions

1. How will the convolution additive self-attention mechanism perform towards American sign language recognition task ?
2. Will CATM outperform in terms of computations efficiency comparted to traditional multi-head self-attention mechanism ?

1.5 Scope of the Research

This study mainly aims at determining what can be modified or add towards new existing architectures of vision transformer systems to obtain a more efficient model which supports for American hand sign recognition using an existing dataset. Furthermore, since the architecture of transformer completely taking a new approach the efficiency measurements are taken into consideration to analyze the performance compared to multi-head model. Since the model only differentiate on the similarity calculation part the architecture does not follow the multiple stages of convolutional additive original architecture. Various parameters applied through the test results and obtaining the highest possible outcome is targeted by the study and will generate comparison between two models.

1.6 Justification of the Research

This study area, which is called “Hand gesture recognition” has not previously involved in many vision transformer base modules since it is a novel approach towards the image processing task that variates from traditional convolutional neural network approaches. Even though one of the latest studies implemented a modified version of vision transformer for successful hand gesture recognition task using American Sign Language data set called HGR-Vit, it was still applied the methodology of multi hit self-attention mechanism to calculate the similarity scores between embeddings of the original image patches. Apart from building a more accurate model towards the ASL recognition task since the computational complexity is higher it cost more time and effort to study the area property. Since the vision transform model performed on a cell ASL recognition successfully reducing the complexity of the calculation within the module can bring so many benefits to those researchers who are currently engaged in building up tools for numerous recognition tasks and later benefit for build special lexicons for private usages.

1.7 Limitations of the Project

- Limited computational capacity :

  Since the vit modules required comparably higher computation capacity to test it may be limited by the limited capabilities of the laptop device that has been used here.

- Time period limitation :

  The research could be limited by the time frame that are given for the allocating into it due to the reason that deep learning modules such as vision Transformers required more time to train models and evaluate.

2. Literature Review

2.1 Introduction to the Research Theme

Emerging technologies such as ML and DL has contributed to the field of “Hand gesture recognition” using static and dynamic methodologies. While static methodologies are used for interpreter the meanings of single gesture symbols the dynamic methodologies are the ones that extract and movement from a given sequence of frames for a given meaning or category. There are numerous CNN models which are developed towards recognizing tasks using American Sign Language data set throughout the past years such as LSTM model, YOLOv5 Classifier. Also, there were some studies that implement vision transformer on this task and get accurate predictions for the correct classifications. However apart from vision Transformers patch connection which are excellent for detection tasks and image processing approach the higher demand of computational power was always a limitation in all the studies were facing before and they are studies for finding a cure for this complex computational approach and the reduce its computational requirements. Some of those studies were able to overcome this an address modification on various parts of multi itself attention mission Transformers to extractor meaning of image patch token mixture approach with comparably lower calculations with minimum drop of accuracy levels.

2.2 Theoretical Explanation about the Key Words in the Topic

Convolutional Additive Attention :

> The word convolutional additive attention describes the special mechanism of the similarity calculation process of vision Transformers that uses convolutional layers and addition between query, key for extract the similarity scores which originally derived from the query, key, value multi-head self-attention mechanism of Transformers.

Vision Transformers :

This indicates the main base model which has been used in the project that is one of the image processing technologies introduced lately. The key difference between this approach and previously used CNN approach is these uses technique called self-attention on details of the images to get the connectivity between pixels but generally demand more computation due to complex matrixes calculations.

2.3 Findings by Other Researchers

One of the latest research Tan, C.K. et al. (2023) has implemented the multi-head self-attention vision Transformers on American Sign Language data sets. They have used the same mission Transformers mechanism with categorical cross entropy those functions for loss calculations and early stopping, adaptive learning rates has been used for optimal results(fig 1)

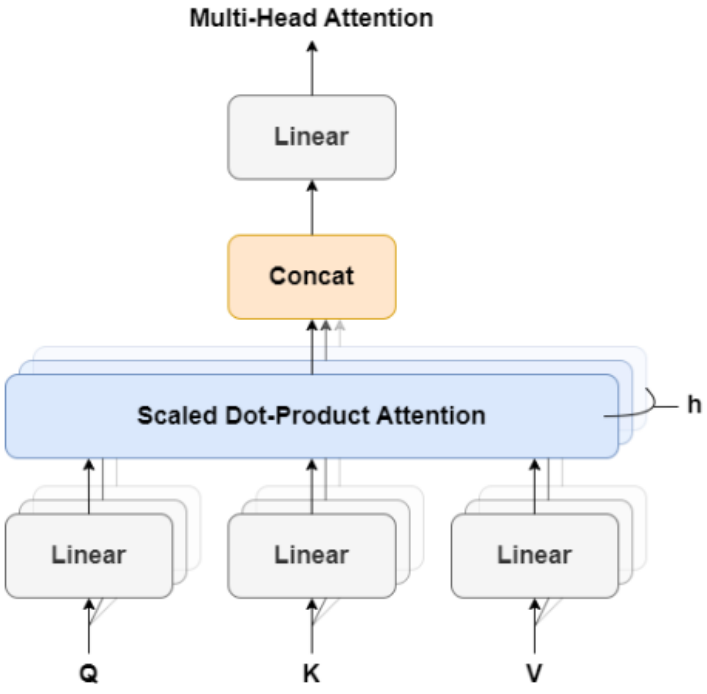


Figure 1: traditional vision transformer architecture with its' loss function, categorical cross entropy

$$L_{CE} = -\sum_{i=1} T_i \log(S_i)$$

They have used three different data sets that are namely American Sign Language (ASL) dataset, ASL with digits dataset, and NUS hand gesture dataset accordingly to evaluate this model called HGR-Vit and obtain training accuracy of 100% and testing accuracy in the range of 99% to 100% for all three data sets. Additionally, they have performed cross validation sets up to five and obtain the same results shown in the figure 2 below.

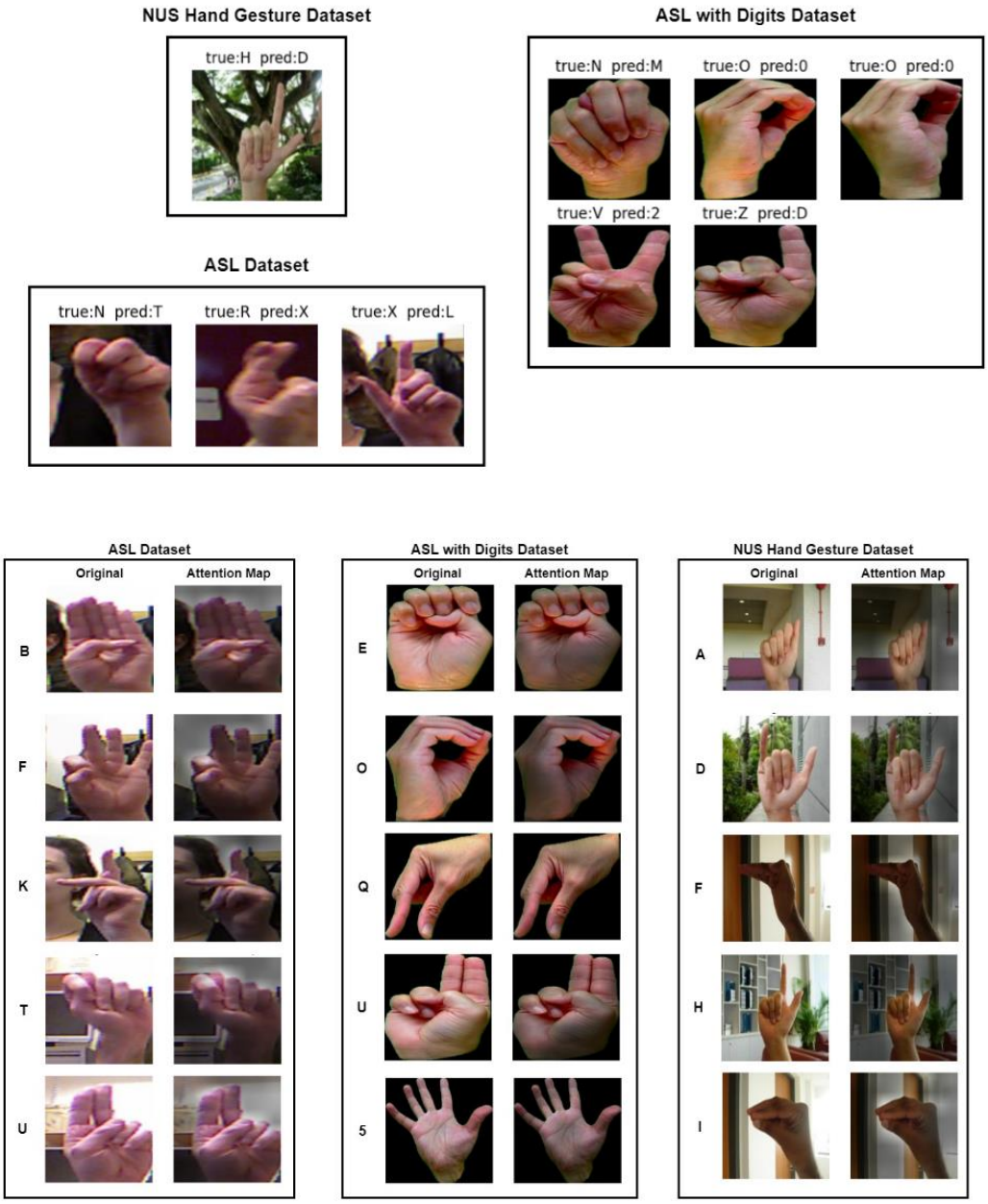


Figure 2 : datasets that are used in Tan, C.K. et al. (2023) and their attention map samples

Another study on optimizing the vision transformer architecture (Zhang, T. *et al.* ,2024) showed that Convolutional Additive Self-Attention which is an efficient approach for traditional Multi-Head Self-Attention (MSA) and more recent advancements such as Swift Self-Attention, which focuses on optimizing Query (Q) and Key (K) calculations in the attention mechanism, the CATM mechanism introduces more generalized efficient similarity function. CATM calculate similarity by summing up the context scores to make less calculation simple unlike traditional self-attention methods that get similarity using complex dot-product calculations, see figure 3. The context score is generated using a context mapping function (Φ), which includes sigmoid-based channel and spatial attention process, thus used for in detail capitulation of both locations and density of objects. Additionally, the CATM mechanism uses convolutional operations to map the input image and thereby to reduce the complexity of O(N) where N denote the number of tokens. This mechanism which combines convolution with similarity calculation awareness helps to preserve more details in image while simplify calculations which helps to address the early feature reduction problem, which was a limitation in earlier works such as Separable Self-Attention. Furthermore, by using sigmoid based attention mechanism this CATM has achieved capability to address inefficiencies which occurred at previous models in figure 3, which were used traditional normalization. This modification not only facilitates enhanced parallelization due to independent sigmoid computation but also allows for effective deployment on mobile devices. As in figure 3, CATM uses attention at every stage and uses convolution which helps to preserve more information compared to models with models that uses attention only at last steps while reducing the complexity of calculations which make it more compatible with mobile devices.

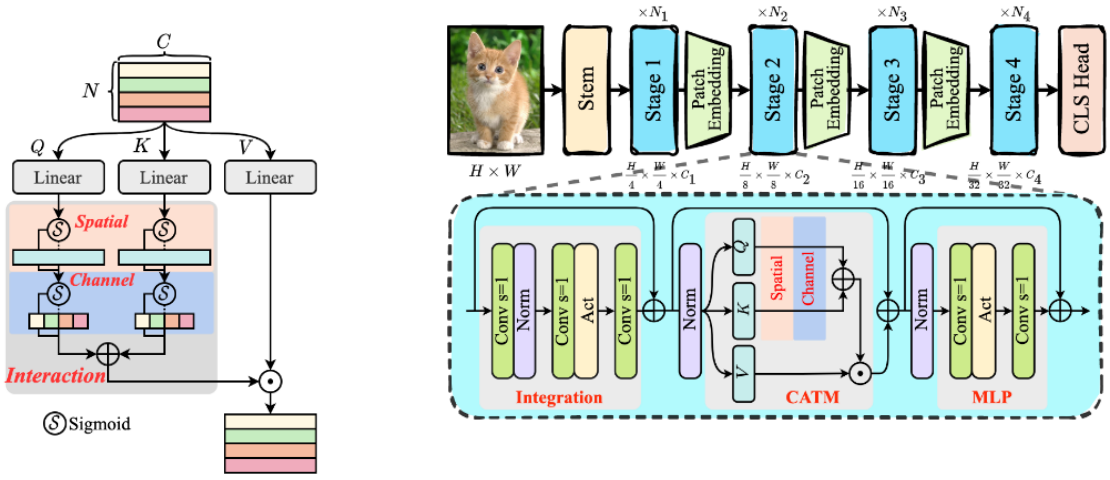


Figure 3 : CATM Architecture

This model achieved Average Precision (AP) metrics over different scales (small, extra-small) and tasks (ImageNet Classifications), alongside their computational efficiency indicated by parameters (Par.) and Flops, see figure 4.

Table 1: **Image classification on ImageNet-1K validation set (Deng et al. 2009).** Throughput results are tested on NVIDIA V100 GPU, Intel Xeon Gold 6248R CPU @ 3.00GHz for ONNX and Apple Neural Engine (ANE) compiled by CoreMLTools on iPhone X IOS 15.2. Note that for throughput, we report data per frame on mobile device, and report the results on GPUs and ONNX with batch size of 64. Our results are shown in bold for all model variants.

Model	Par.↓ (M)	Flops↓ (M)	Throughput(images/s)↑ GPU	ONNX	ANE	Top-1↑ (%)	Model	Par.↓ (M)	Flops↓ (M)	Throughput(images/s)↑ GPU	ONNX	ANE	Top-1↑ (%)
SpectFormer-T	8.88	1688	2389	-	-	76.9	PoolFormer-S24	21.35	3394	1258	36.06	3.26	80.3
MobileOne-S1	4.76	830	4323	1366.80	20.85	77.4	SpectFormer-XS	19.61	3765	1581	-	-	80.2
FAT-B0	4.49	710	1688	50.97	6.00	77.6	AFNO-ViT-S/4	14.88	2840	1245	-	-	80.9
MobileNetv3-L-1.0	4.18	215	6303	226.38	36.70	75.2	MobileOne-S4	14.82	2987	1468	125.20	9.70	79.4
EdgeViT-XXS	4.07	546	2438	107.16	10.31	74.4	PVTv2-B1	14.00	2034	1385	21.91	3.41	78.7
EfficientFormerV2-S0	3.59	400	1008	73.83	19.13	75.7	EfficientViT-M5	12.47	525	5104	277.10	-	77.1
SwiftFormer-XS	3.47	608	3184	132.69	17.11	75.7	SwiftFormer-L1	12.05	1604	2047	80.62	9.93	79.8
EMO-2M	2.38	425	2672	80.11	19.07	75.1	GC ViT-XXT	11.94	1939	951	41.45	2.76	79.9
MobileViT-XS	2.31	706	2893	129.13	17.90	75.7	MobileViTv2-1.5	10.56	3151	1526	58.38	6.23	80.4
CAS-ViT-XS	**3.20**	**560**	**3251**	**115.30**	**14.63**	**77.5**	**CAS-ViT-M**	**12.42**	**1887**	**1566**	**50.39**	**5.49**	**81.4**
SDT-8-384	16.43	18802	-	-	-	72.3	ViT-B/16	86.42	16864	705	18.27	0.93	77.9
PoolFormer-S12	11.90	1813	2428	75.51	6.33	77.2	ResNeXt101-64x4d	83.46	15585	606	36.84	2.75	79.6
MobileOne-S3	10.07	1902	2899	652.73	16.16	80.0	PoolFormer-S36	32.80	14620	853	17.29	2.16	81.4
EfficientViT-M4	8.80	301	6975	404.03	-	74.3	SpectFormer-S	32.03	6197	966	-	-	81.7
FAT-B1	7.81	1186	1290	37.77	4.27	80.1	SDT-8-512	29.18	33238	-	-	-	74.6
EdgeViT-XS	6.78	1123	2044	85.17	6.77	77.5	SwiftFormer-L3	28.48	4021	1309	42.23	6.4	83.0
EfficientFormerV2-S1	6.18	666	881	55.51	14.33	79.0	Swin-T	28.27	4372	977	-	-	81.3
SwiftFormer-S	6.09	991	2607	108.98	12.90	78.5	PVT-S	24.10	3687	933	40.62	3.08	79.8
EMO-5M	5.11	883	1971	53.61	10.43	78.4	Twins-SVT-S	24.06	2821	1202	46.45	-	81.7
MobileViTv2-1.0	4.88	1412	2429	91.40	11.52	78.1	Wave-ViT-S	22.69	4391	653	-	-	82.7
CAS-ViT-S	**5.76**	**932**	**2151**	**73.06**	**9.92**	**80.2**	**CAS-ViT-T**	**21.76**	**3597**	**1084**	**38.67**	**3.83**	**82.3**

Figure 4 : CAS-ViT result evaluations

2.4 The Research Gap

Even though this area is hugely studied by the previous researchers through numerous studies the implementation of vision transformers been limited and modification of the mathematical computational complexity of these modules have not apply to the ASL hand gestures to study how it will be effect on classifications and how it will perform based on efficiency measurements such as flops, runtime compared with its top accuracy acquired. Since CNN architectures add less complex approach towards the various pattern recognition adding the vision transformer base self-attention mechanism combined with convolutional based approach is still a gap that should study in order to achieve efficient models with high accurate results.

3. Methodology

3.1 Introduction

Data collection methodologies, data preprocessing, model implementation details and proposed architecture components towards the model building will be discussed throughout this chapter.

3.2 Type of Data to be Collected and Data Sources

Data will be taken from the "American Sign Language Dataset" (Yu, F. ,2020) for it includes 36 classes which describes signs for English alphabet letters and 0 to 9 numeric. Data will be primarily downloaded from there Kaggle source this page since they are major distribution and publication of various data collections.

3.3 Data Collection Tools and Plan

Data will be collected only for instance segmentation labels and since there is a huge collection of data around 1.8 TB data withing the database and there is a limitation of external hardware the diction of only take a proportion of collection were taken by the date this proposal submitted. (https://www.kaggle.com/datasets/ayuraj/asl-dataset)

3.4 Methods of data analysis

Here mainly included a comparison between two models which are multi head self-attention based traditional vision Transformers and convolutional additive similarity calculation-based vision Transformers. The multi head vision transformer with dot product-based similarity calculation took from the keras' version of vision Transformers published on https://keras.io/examples/vision/image_classification_with_vision_transformer/ and for the similarity calculation part the convolutional additive phase was joined and created another modified version of vision Transformer to process the images and get the results.

The traditional version was run separately for accuracy measurements in total parameter counts, Floating Points operation per second (Flops) and counted the total runtime, inference time after each epoch. After that the modified version which includes convolutional additive similarity model was simply carried through the same process and got the calculations separately. The measurements were late analyst through the same graphs which includes both models' performance on a same scale to compare between both and get a performance comparison.

4.0 Data Analysis and Findings and Interpretation

4.1 Data Preprocessing

First the hand gesture images, and their labels were loaded from the original directories and split them into training(80%), test(20%) and validation(10%) sets for feeding into the model(fig 5). Additionally, the images and their corresponding labels shuffled using scikit learn utility function to prevent the overfitting to the module.

```
Found classes: ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f',
Loaded 2515 images from 36 classes.
Train set: 1810, Validation set: 202, Test set: 503
```

Figure 5 : dataset details

Then labels are assigned for each category with label encoder functions for the data must feed into model as arrays which are created using indexes assigned to each category respectively.

```
Label mappings:
0: 0
1: 1
2: 2
3: 3
4: 4
5: 5
6: 6
7: 7
8: 8
9: 9
10: a
11: b
12: c
13: d
14: e
15: f
16: g
17: h
18: i
19: j
20: k
21: l
22: m
23: n
```

Figure 6 : encoded labels for each class

Also, images are loaded in shape of (200, 200, 3) while reading from the original paths and used for further processing.

As for the data augmentation, multiple stacked layers were created on top of which model to first normalize the image using the mean and variance of pixel distribution, they'll further reduce the image size using keras resizer and then randomly flipped using 'horizontal' way and random flip and random show accordingly with the rotation and zoom factor of 0.2 for width and height accordingly.

```
data_augmentation = keras.Sequential(
    [
        layers.Normalization(),
        layers.Resizing(image_size, image_size),
        layers.RandomFlip("horizontal"),
        layers.RandomRotation(factor=0.02),
        layers.RandomZoom(height_factor=0.2, width_factor=0.2),
    ],
    name="data_augmentation",
)
# Compute the mean and the variance of the training data for normalization.
data_augmentation.layers[0].adapt(train_x)
```

Those augmented images then follow the part where they partitioning into patches based on the image size and patch sizes. Since the images were converted to optimal number of heigh and width which is (128,128) during data augmentation part the patch size were fixed into (16,16) pixels which makes them total 64 patches for an image altogether and 768 elements considering the RGB layers; see fig 7.

```
Image size: 128 X 128
Patch size: 16 X 16
Patches per image: 64
Elements per patch: 768
```

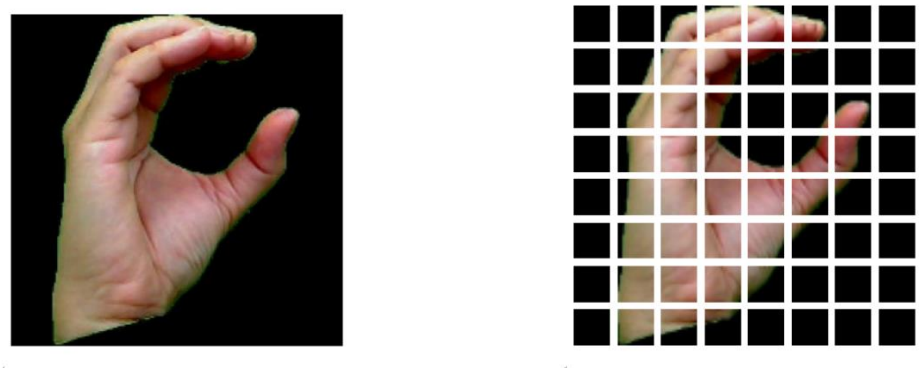


Figure 7 : Before and After Partitioning to Patches

Each patch has projected into 64-dimension embeddings which reduce original linear projections 256 vector embeddings to reduce computation complexity with comparably lesser data loss.

4.2 Multi-Head Traditional Vision Transformer Building

While implementing the vision transform model here used Dosovitskiy et al. (2020) architecture which were published on https://keras.io/examples/vision/image_classification_with_vision_transformer/ and the architecture follows the following architecture; see fig 8 as a summary. It includes partitioning position embedding simulate calculations and finally classify through multilayer perception.

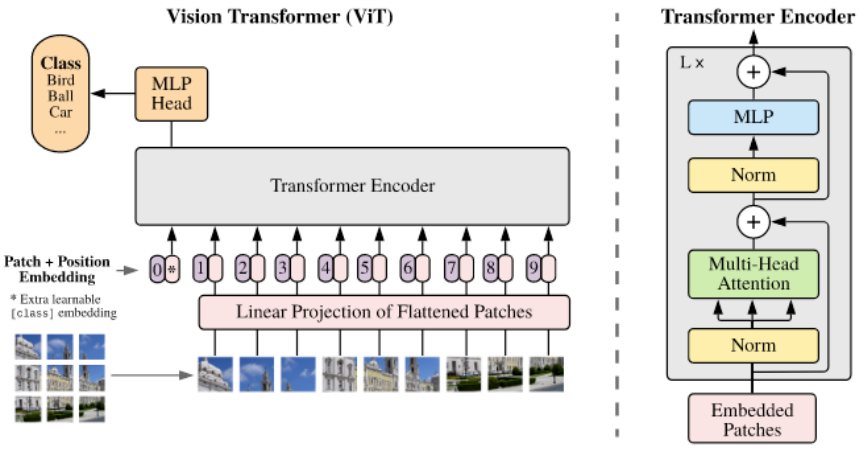


Figure 8 : architecture in Dosovitskiy et al. (2020) which was published in keras

1. **Patch Embedding**: A 2D image $x \in \mathbb{R}^{H \times W \times C}$ is divided into patches of size $P \times P$, resulting in $N = \frac{H \times W}{P^2}$ patches. Each patch is flattened and projected to a dimension D using matrix E:

$$z_0 = [x_{\text{class}}; x_1^p E; x_2^p E; \ldots ; x_N^p E] + E_{\text{pos}}$$

where $E_{\text{pos}} \in \mathbb{R}^{(N+1) \times D}$ adds positional information.

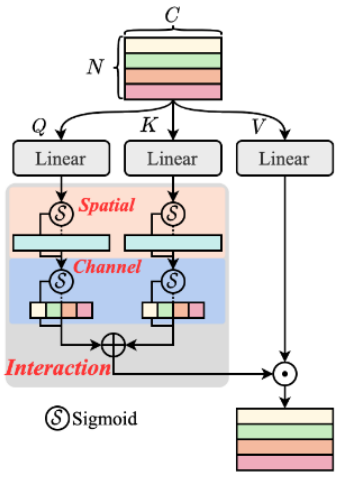
2. **Transformer Encoder**: The sequence z is processed through L layers of multi-head self-attention (MSA) and MLP blocks with Layer Normalization (LN) and residual connections:
   - **MSA**: $z'_\ell = \text{MSA}(\text{LN}(z_{\ell-1})) + z_{\ell-1}$
   - **MLP**: $z_\ell = \text{MLP}(\text{LN}(z'_\ell)) + z'_\ell$
3. **Classification Output**: After the final Transformer layer, the class embedding z_0^L is normalized:

$$y = \text{LN}(z_0^L)$$

During fine-tuning, y is passed to a classification head for downstream tasks.

4.3 Convolutional Additive Vision Transformer Building

While performing the same steps as the previous for the vit model this part includes only the modification on similarity score calculation according to the sigmoid based additive mechanism. This part has directly extracted from the paper where they have implemented sigmoid based additive approach to calculate the similarities scores between query and key pairs (fig 9).



(d) Conv Additive Self-Attention

Figure 9 : modified similarity calculation mechanism called convolutional additive in Zhang, T. et al. (2024)

$$Sim\,(\mathbf{Q}, \mathbf{K}) = \Phi(\mathbf{Q}) + \Phi(\mathbf{K})\ s.t.\ \Phi(\mathbf{Q})$$

4.4 Parameter Tuning in Traditional Vision Transformer

LR = learning rate, WD = weight decay, BS = batch size

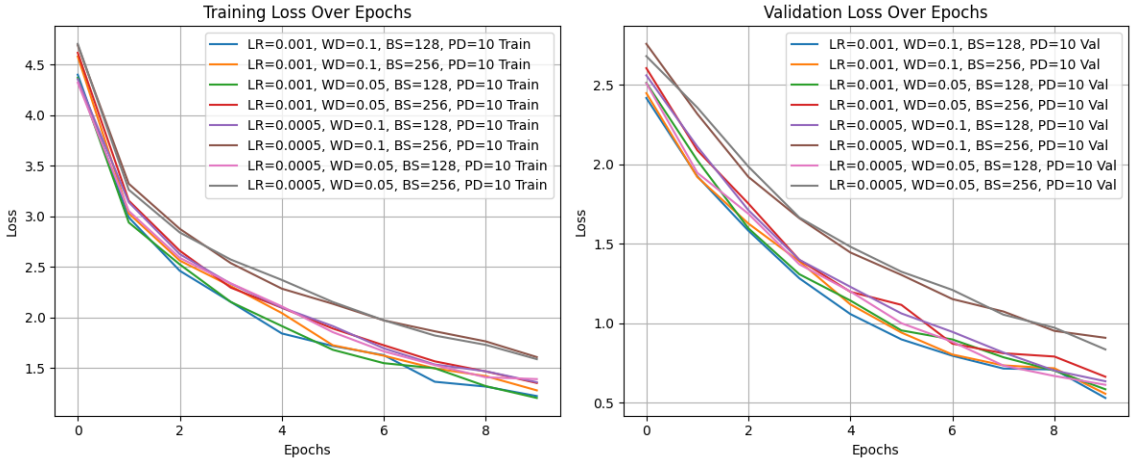


Figure 10 : training and validation losses in Dosovitskiy et al. (2020) ViT on ASL

According to the training and validation loss over the epochs the blue, green and orange lines have highest decline over epochs which are giving the parameter combinations of LR, WD, BS ; (0.001, 0.1, 128, 10), (0.001, 0.05, 128, 10) and (0.001, 0.1, 256, 10) respectively.

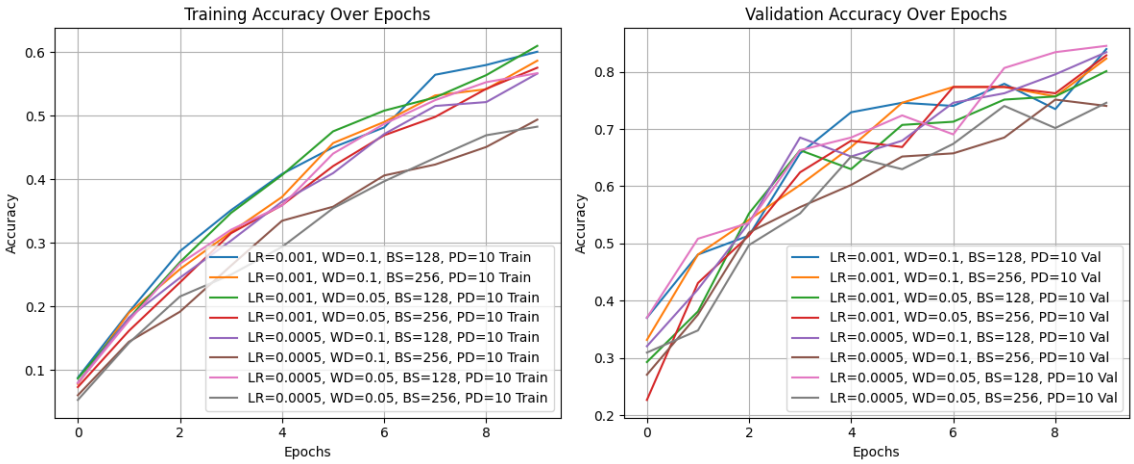


Figure 11 : training and validation accuracies in Dosovitskiy et al. (2020) ViT on ASL

As a reflection of loss and increasement of accuracy over the epochs blue, green and orange lines have marked as highest accuracy obtainers here in the figure 11.

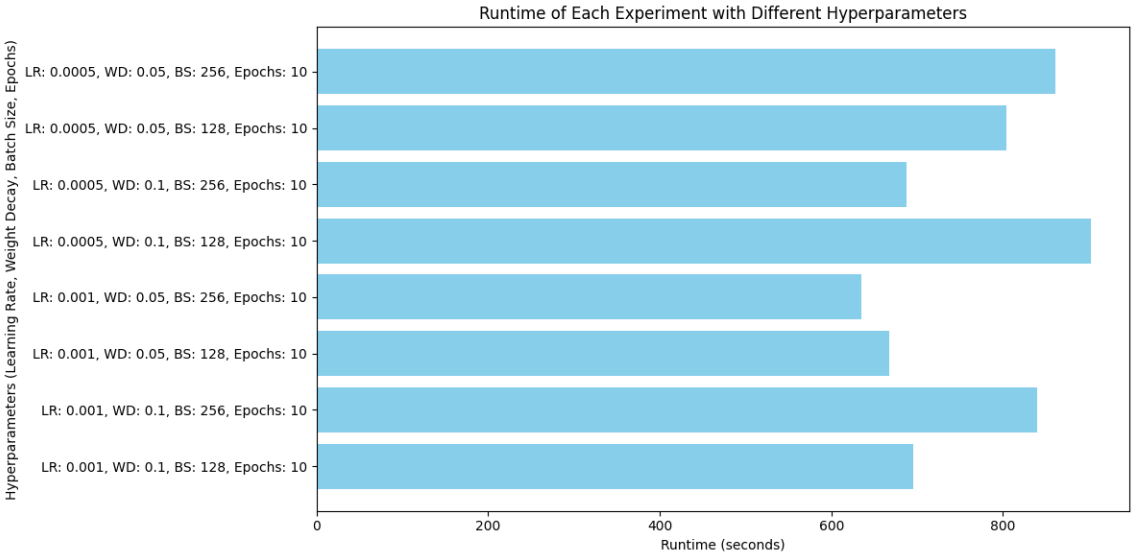


Figure 12 : training runtimes for each parameter combinations in Dosovitskiy et al. (2020) ViT on ASL

Considering the highest accuracy and lowest accuracy obtainer hyperparameters (0.001, 0.1, 256, 10) combination obtained the lowest runtime according to the figure 12. For the further evaluation of model learning rate sat into 0.001, weight decay sat to 0.1 and batch size was set to 256 (Epochs were fixed at 10 for evaluate parameters first at this step).

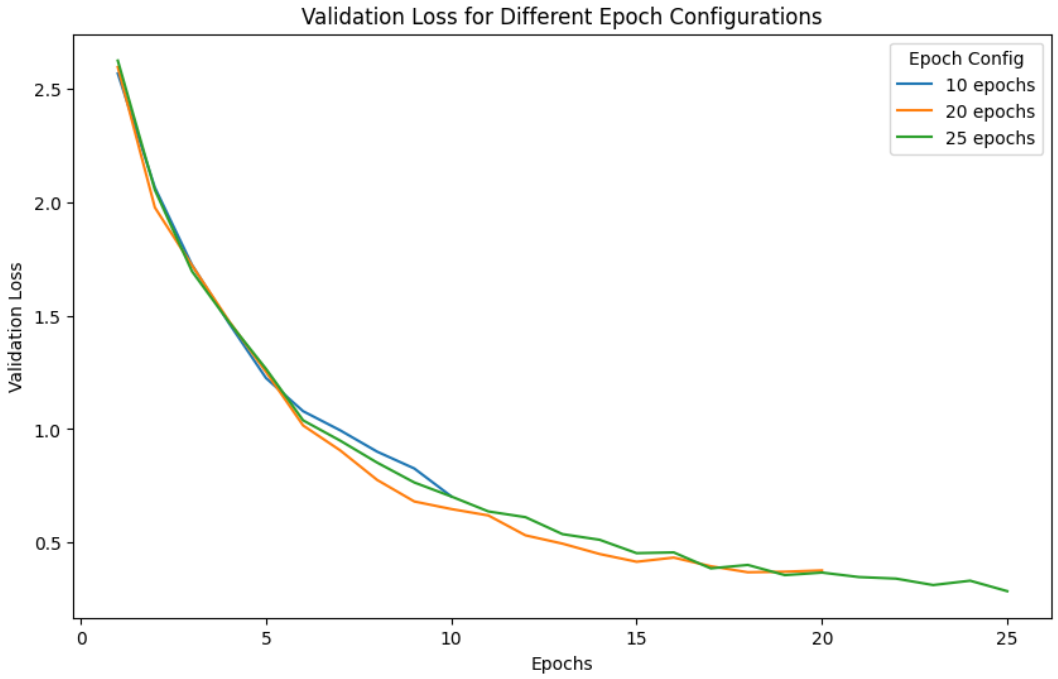


Figure 13 : validation loss over epochs in Dosovitskiy et al. (2020) ViT on ASL

```
Test accuracy: 92.25%
Test top 5 accuracy: 99.6%
Best number of epochs based on validation loss: 25
```

Epochs were limited to 25 maximums due to the time consumption and according to the evaluation traditional Vit get good accuracy at maximum epochs which was 25. Under the assumption that limitation of computation capability was not sufficient to tune the number of epochs, both models were decided to run on 25 epochs to evaluate.

4.5 Parameter Tuning in Convolution Additive Modified Vit

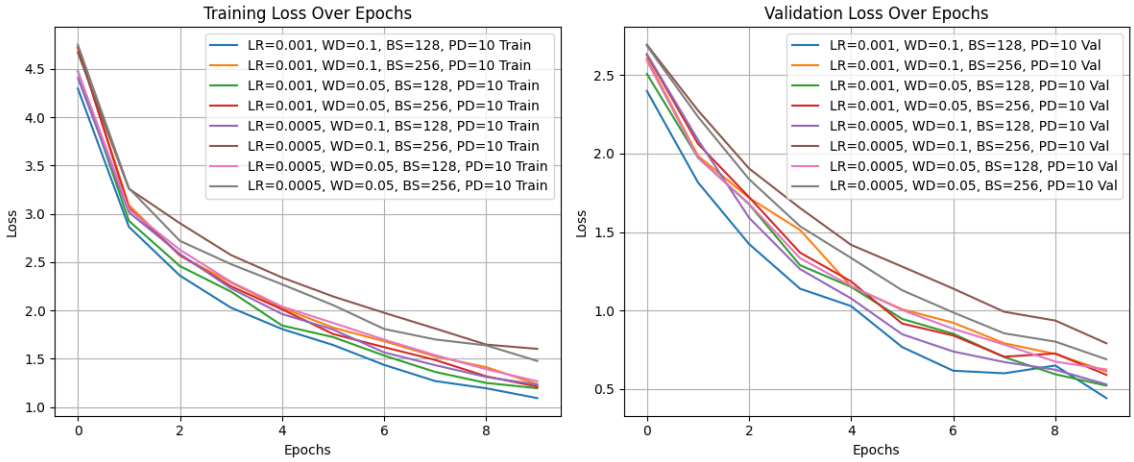


Figure 14 : training and validation losses in CAT-ViT on ASL

According to the training and validation loss of Con-vit over the epochs the blue, green and purple lines have highest decline over epochs which are giving the parameter combinations of LR, WD, BS ; (0.001, 0.1, 128, 10), (0.001, 0.05, 128, 10) and (0.0005, 0.1, 128, 10) respectively.

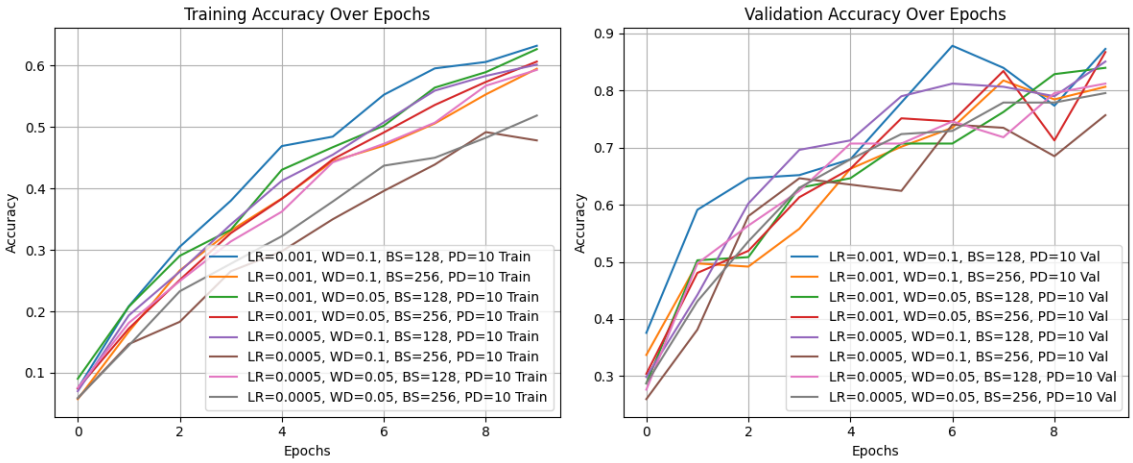


Figure 15 : training and validation accuracies in CAT-ViT on ASL

Again, the same three lines with their parameters has obtain highest values over the epochs here in terms of accuracies; see fig 15.

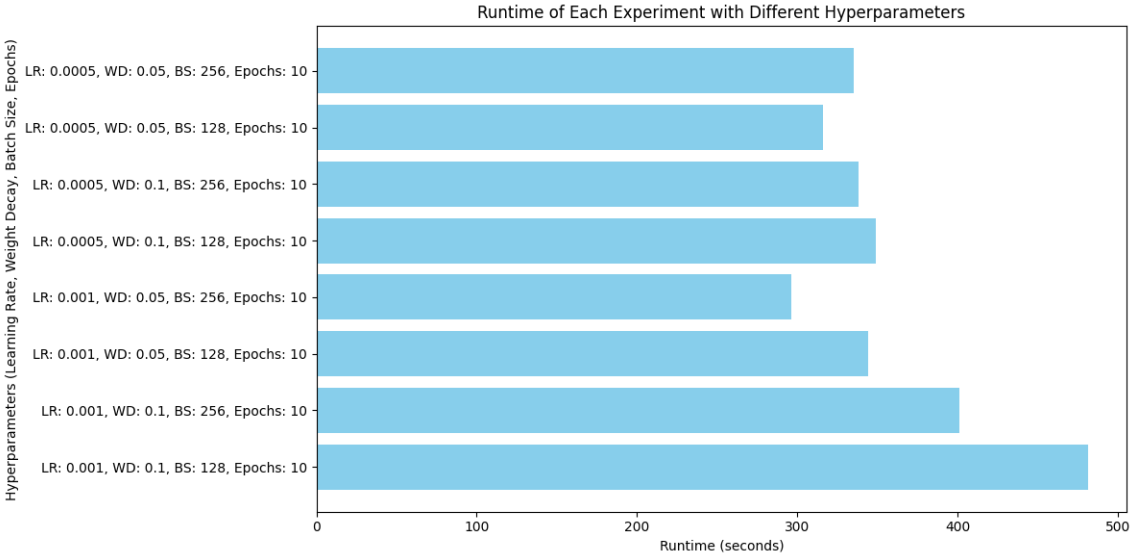


Figure 16 : training runtimes for each parameter combinations in CAT-ViT on ASL

Even though the lowest accuracy was obtained by the learning rate of 0.001 weight decay of 0.05 and batch size of 256 she is it was not among the best performing models previously, the next best runtime performance combination which is marked best performance chosen as the combination of learning rate of 0.001 weight decay of 0.05 and batch size of 128. Nevertheless, these all combinations in Con-Vit have less runtime overall

compared to traditional dot product-based models (Epochs were fixed at 10 for evaluate parameters first at this step).

4.6 Comparison of Models

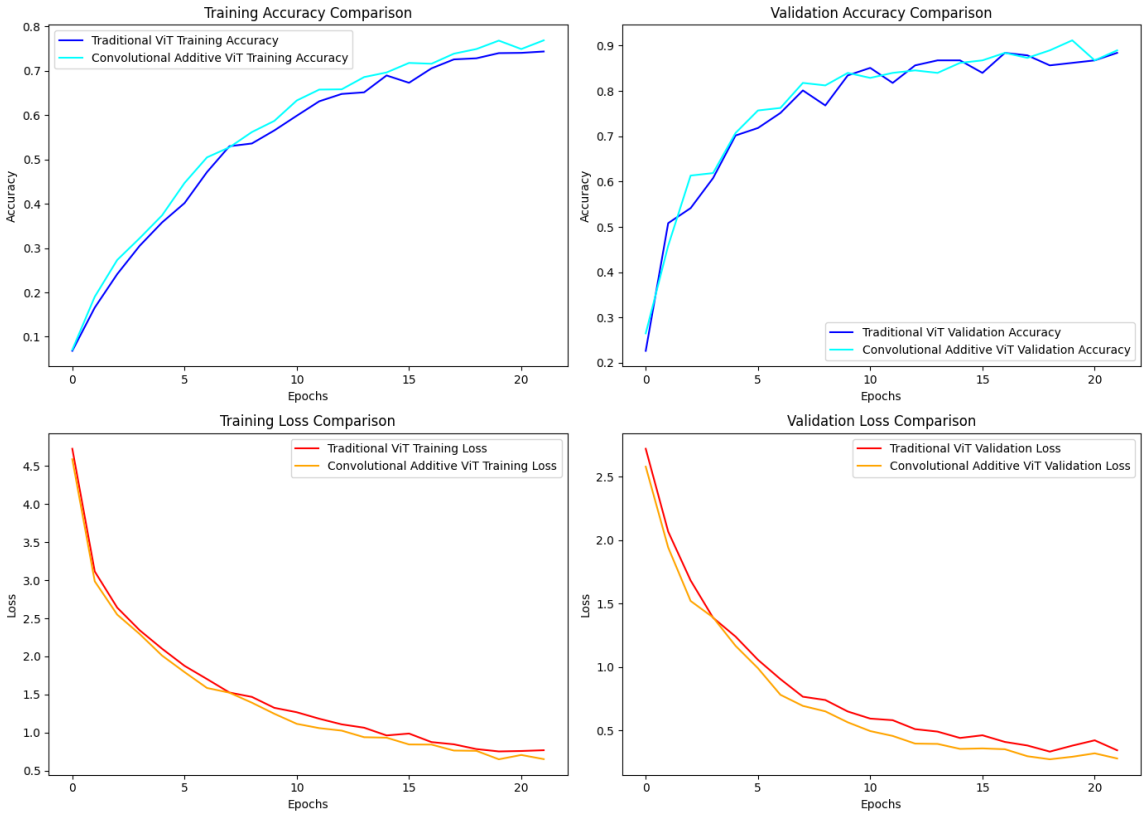


Figure 17 : comparison of training loss and accuracy and validation test and accuracy on ASL for both models

Convolution additive attention-based model has slightly outperformed in both training and validation tasks as shown in the figure 17.

Metric	Traditional ViT	Convolutional Additive ViT
Learning Rate	0.001	0.001
Weight Decay	0.1	0.05
Batch Size	256.0	256.0
Epochs	25.0	25.0
Total Parameters	11509995.0	10913515.0
FLOP Count (M)	27.05	24.66
Training Runtime (s)	1388.64	717.21
Test Loss	0.3181	0.2607
Test Accuracy (%)	88.07	90.46
Test Top-5 Accuracy (%)	100.0	99.8
Inference Time per Sample (ms)	46.67	18.02

Figure 18 : all the accuracy and efficiency matrixes on ASL for both models

As the above figure 18 indicates after fixing the parameters accordingly under the fix size of epochs into 25 the total parameter count, training runtime and inference time per sample took into consideration regarding the model performance efficiency matrixes and test loss best accuracy and top five accuracy indicates accuracy of the model performance. Between two models there is not much of accuracy difference between two models since they had obtained 88.07% and 90.46% accuracy respectively. And top 5 accuracy is 100 and 99.8 for the models vit and convolution vit. Convolution vit has higher performance in total parameters, training runtime and inference time since it has lower parameters which is 10 M compared to 11 M in traditional vit and training runtime and inference time is nearly half from the traditional vit model. Floating points operation per seconds (Flops) are 27 M and 24 M for models accordingly.

Classification Report:

	precision	recall	f1-score	support
0	0.9000	0.9000	0.9000	10
1	0.8824	1.0000	0.9375	15
2	0.6522	0.9375	0.7692	16
3	0.9375	0.8824	0.9091	17
4	0.8750	0.7778	0.8235	9
5	0.9231	0.9231	0.9231	13
6	0.7143	0.2941	0.4167	17
7	1.0000	0.8947	0.9444	19
8	0.8889	0.9412	0.9143	17
9	1.0000	0.8571	0.9231	14
a	1.0000	0.9000	0.9474	10
b	0.9167	1.0000	0.9565	11
c	1.0000	0.9375	0.9677	16
d	0.8571	0.8000	0.8276	15
e	1.0000	1.0000	1.0000	13
f	0.9412	0.9412	0.9412	17
g	1.0000	0.9231	0.9600	13
h	0.9474	1.0000	0.9730	18
i	1.0000	1.0000	1.0000	11
j	1.0000	1.0000	1.0000	11
k	0.9500	0.9500	0.9500	20
l	0.9444	1.0000	0.9714	17
m	0.7143	0.7143	0.7143	7
n	0.8947	1.0000	0.9444	17
o	0.8571	0.9231	0.8889	13
p	1.0000	1.0000	1.0000	16
q	1.0000	1.0000	1.0000	10
r	1.0000	0.9474	0.9730	19
s	1.0000	0.8889	0.9412	9
t	1.0000	1.0000	1.0000	18
u	0.8462	1.0000	0.9167	11
v	1.0000	0.2000	0.3333	10
w	0.4800	0.9231	0.6316	13
x	1.0000	1.0000	1.0000	19
y	1.0000	1.0000	1.0000	11
z	0.7273	0.7273	0.7273	11
accuracy			0.9026	503
macro avg	0.9125	0.8940	0.8896	503
weighted avg	0.9159	0.9026	0.8973	503

Figure 19

Looking into classification details of traditional Vit has achieved full score in precision, recall, f1-score. The model more precisely captures alphabetic letter signs than digit signs. Letter M, Z, 2, 6 have shown a drastically drop in all the three measures compared to other signs. These could be due to the limited test sets and alike signs. Overall model has achieved accuracy of 90% and a weighted average of 89% (fig 19).

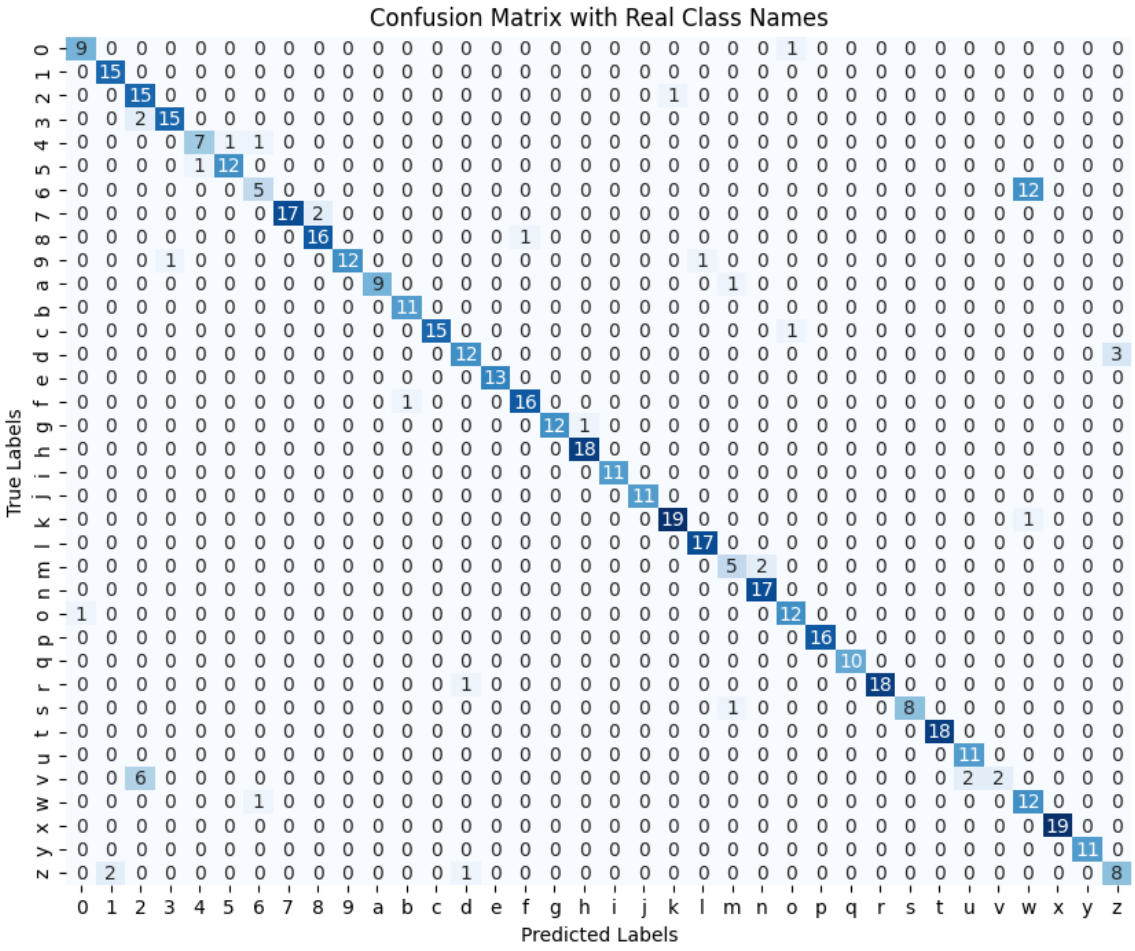


Figure 20 : confusion matrix of Dosovitskiy et al. (2020) ViT on ASL

Although most of the classes were predicted correctly the confusion matrix shows the confusions between alike gestures which belongs to letter V with 2, W and 6 specially. Between the confusions W and 6 shows a strong confusion since it has 12 incorrect predictions.

Classification Report:

	precision	recall	f1-score	support
0	1.0000	0.8000	0.8889	10
1	1.0000	0.9333	0.9655	15
2	1.0000	0.7500	0.8571	16
3	1.0000	0.8824	0.9375	17
4	0.8750	0.7778	0.8235	9
5	0.9231	0.9231	0.9231	13
6	0.7273	0.4706	0.5714	17
7	0.9444	0.8947	0.9189	19
8	0.8947	1.0000	0.9444	17
9	0.8667	0.9286	0.8966	14
a	0.7143	1.0000	0.8333	10
b	0.8462	1.0000	0.9167	11
c	1.0000	1.0000	1.0000	16
d	1.0000	0.7333	0.8462	15
e	1.0000	1.0000	1.0000	13
f	1.0000	0.8824	0.9375	17
g	1.0000	0.4615	0.6316	13
h	0.7200	1.0000	0.8372	18
i	1.0000	1.0000	1.0000	11
j	1.0000	1.0000	1.0000	11
k	1.0000	0.9500	0.9744	20
l	1.0000	1.0000	1.0000	17
m	0.7000	1.0000	0.8235	7
n	1.0000	0.8824	0.9375	17
o	0.8667	1.0000	0.9286	13
p	1.0000	1.0000	1.0000	16
q	1.0000	1.0000	1.0000	10
r	1.0000	0.7368	0.8485	19
s	1.0000	0.7778	0.8750	9
t	1.0000	0.8333	0.9091	18
u	0.6471	1.0000	0.7857	11
v	0.6667	1.0000	0.8000	10
w	0.5500	0.8462	0.6667	13
x	1.0000	1.0000	1.0000	19
y	1.0000	1.0000	1.0000	11
z	0.7333	1.0000	0.8462	11
accuracy			0.8966	503
macro avg	0.9076	0.9018	0.8923	503
weighted avg	0.9187	0.8966	0.8962	503

Figure 21

The modified convolutional additive approach has been able to much more prediction classes with high precision and recall compared to the traditional model. There are no significant drops in measurements regarding all three classes and the lowest drops are related to letter W and 6 only. Overall, this model has achieved accuracy of 89.66% on testing data with weighted average of 89.62%.

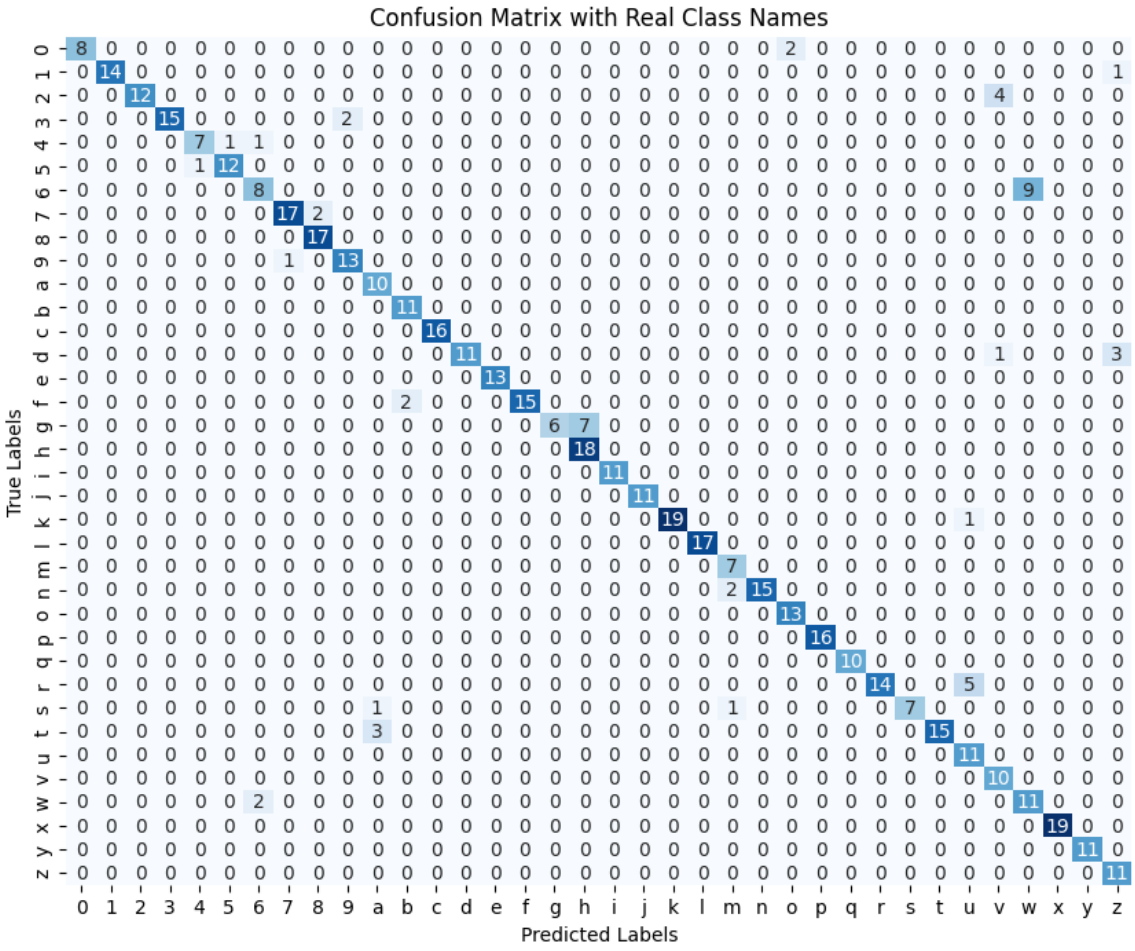


Figure 22 : confusion matrix of CAT-ViT on ASL

The conventional model was also performed a detection same as the previous model since it has also confused letter W with digit 6 on alike signs. Even though the confused count is lower it has shown some significant confusions with other signs like letter U and R, A and T, V and 2. Unlike the traditional Vit model this indicates lowering testing accuracy with unable to differentiate details in patch embeddings into some degree.

5. Discussion And Conclusions

5.1 Discussion

Traditional multi hit vision transformer model and developed convolutional additive similarity-based vision transform model have built using Kera's version of code implementation which was original derived from Alexander's papers. ASL data set where preprocessed and feed into evaluate the relevant hyperparameters for optimal results in both accuracy and efficiency and fix them into final module evaluations. Throughout the parameter evaluation the epoch size was fixed into 10 and evaluate the learning rate, weight decay and batch size which performs highest accuracy in terms of lost and top 5 accuracy. Total runtime of each parameter combination was taken into consideration for efficiency evaluation and using both accuracy and efficiency matrixes best suitable parameters were fixed.

Then the number of epochs the evaluator then limited according to the computational consuming time and further evaluate the two models. Even though there were no significant accuracy difference between the two models the modified convolutional vision transformer has able to perform slightly better accuracy at training and validation phases. Convolutional vision transformer has obtained higher performance at the testing phase also with 90.46% compared to the traditional vision transformer with 88.07% accuracy at the testing phase. Both models had capability to reach up to 100% top five accuracy over 25 epochs. Discussing on the efficient matrixes of both model convolutional vision transformer yes again overpowered the traditional vision transformer with its smaller number of parameters which is 10.91 million compared to the 11.5 million parameters in traditional Vit. Why is that flops count of convolutional Vit limited to 24 million and traditional Vit has 27 million has remarked the performance efficiency on convolutional Vit. Training runtime of traditional vision transformer Bing 1388 compared to the modified convolutional Vit with its 717 also display efficiency of convolutional model. Additionally, inference time per sample of traditional Vit it's been 46 and convolutional it is being 18 make it convolutional model more suitable for real time analysis compared to the traditional vision transformer. All the measurements that have been taken into consideration for the model performance

indicates that convolutional module perform better at recognizing American hand sign language eat with minimum amount of computational time and highest accuracy. Even though the traditional model has performed nearly even the reduction of computational cost can be more beneficial towards and gesture recognition. As for the final evaluation both models have performed better towards predicting 36 different classes although both how confused some of the signs which are more alike than the other signs such as V and 2, W and 6 according to confusion matrixes.

All the evidence leads towards modern way of optimizing the vision transformer for classification tasks using convolutions and their superior performance in various matrixes. Even though some matrixes shows that the higher computational complexity can leads to better results in capturing signs that are alike and differentiate between them, the previously implemented convolutional additive mechanism shows nearly equal performance with far less computation complexity.

5.2 Recommendations

Since this research was conducted on finding American sign language adaptability into convolutional Vit model there are other dataset that this can be evaluate on which are distributed into various lexicons and regions across the world. Also, video domain includes hand gesture recognition task that can approach through this and developed efficient models for such tasks. Under the limitations of device environment which were those models evaluated there was huge barrier to reduce computation consumption such as epoch count, embedding dimensions, patch sizes etc. Although, they may not be affected by the higher computation capacity since the time consumption is too high if these performances were evaluated under an upgraded environment more accurately and precious result can be obtained. Finally, the further modifications on the tested convolution Vit mechanism can lead to more less complex performance and should be further implemented.

5.3 Conclusions

It can be mainly concluded that between the two types of models we have performed which were traditional multi-head self-attention and convolutional additive attention, convolutional additive attention model has achieved greater success in hand gesture recognition task yielding higher testing accuracy, lower test loss meanwhile maintaining lower flop count, parameter count and very lower training time with inference time to predictions.

Also considering the inference time and flop count this modification on attention is very suitable for real time applications and mobile applications same as they have mentioned in the original paper. It makes it easier to training on specific segments of image classification tasks since it has very low training time.

Apart from training on limited number of epochs and fixed dimension size it can be concluded that training model more at higher dimensional patch embeddings and epochs can result in more accurate and efficient model preventing underfitting and improve similar sign distinguishing.

6. References

Zhang, T. *et al.* (2024) *CAS-VIT: Convolutional additive self-attention vision transformers for efficient mobile applications*, *arXiv.org*. Available at: https://arxiv.org/abs/2408.03703 (Accessed: 02 November 2024).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. and Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv:2010.11929 [cs]*. [online] Available at: https://arxiv.org/abs/2010.11929.

Tan, C.K., Lim, K.M., Chang, R.K.Y., Lee, C.P. and Alqahtani, A. (2023). HGR-ViT: Hand Gesture Recognition with Vision Transformer. *Sensors*, [online] 23(12), p.5555. doi:https://doi.org/10.3390/s23125555.