Gesture Classification - Documentation

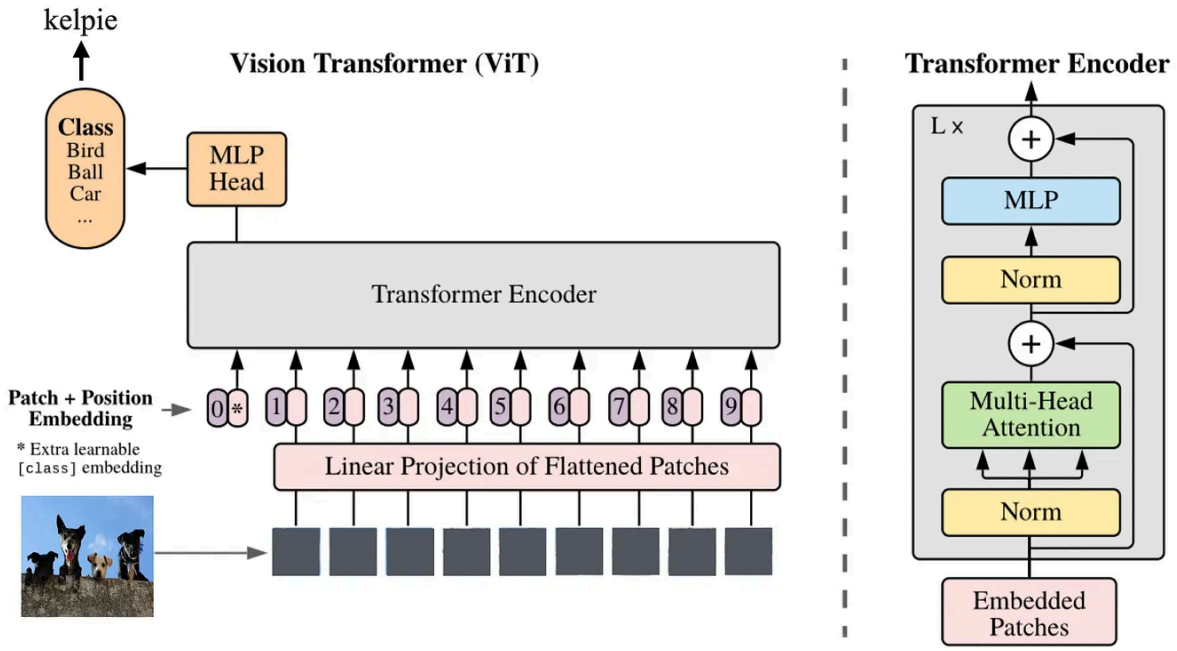
1- Architectures used in the project

Architecture	Main Idea	Step-by-Step Explanation
Vision Transformer (ViT)	Uses transformer models instead of convolution layers by treating images as sequences	1. The input image is divided into fixed-size patches. 2. Each patch is flattened and converted into a vector using a linear embedding. 3. Positional embeddings are added to keep spatial information. 4. The patch embeddings are passed through Transformer Encoder layers. 5. Each encoder uses self-attention to learn relationships between patches. 6. A special classification token collects global image information. 7. The output is passed to a fully connected layer for gesture classification.
ResNet	Uses residual (skip) connections to allow very deep networks	1. The input image passes through initial convolution and pooling layers. 2. Feature maps enter residual blocks. 3. Each block contains convolution, batch normalization, and activation layers. 4. Skip connections add the block input directly to its output. 5. This helps prevent vanishing gradients during training. 6. Deeper layers extract complex gesture features. 7. Global average pooling and a fully connected layer produce the final output.

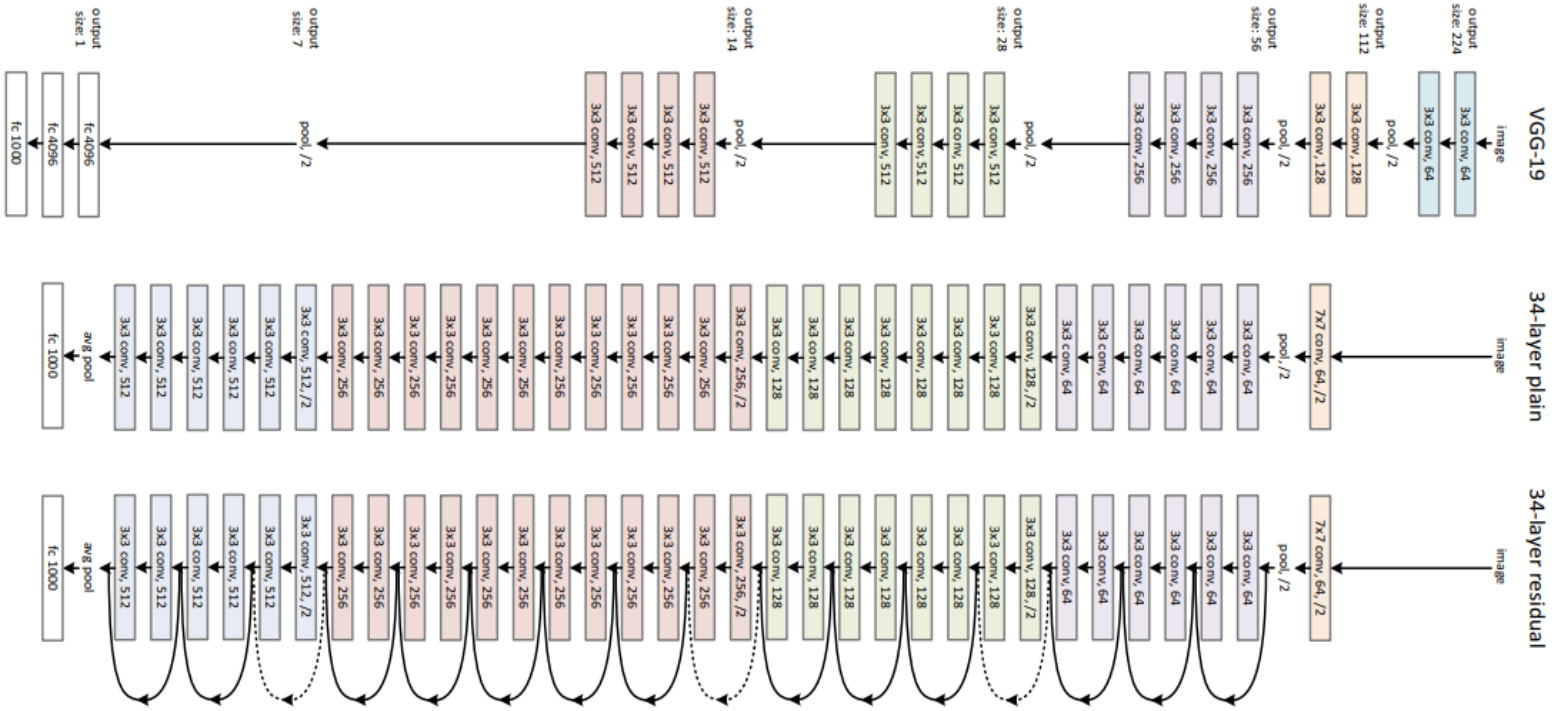
VGG19	Deep CNN with simple and uniform structure using small filters	1. The input image is processed using 3×3 convolution layers. 2. ReLU activation is applied after each convolution. 3. Max pooling layers reduce spatial dimensions. 4. The network depth increases gradually up to 19 layers. 5. Extracted features are flattened into a vector. 6. Fully connected layers learn high-level patterns. 7. A softmax layer performs gesture classification.
Inception V1	Uses parallel convolutions to capture features at multiple scales	1. The input image passes through initial convolution layers. 2. Inception modules apply multiple filters in parallel (1×1, 3×3, 5×5). 3. A max-pooling operation also runs in parallel. 4. Outputs from all paths are concatenated. 5. 1×1 convolutions reduce computational cost. 6. Multiple inception modules are stacked together. 7. Global average pooling and softmax produce the final prediction.

2- Graphs explaining the architecture

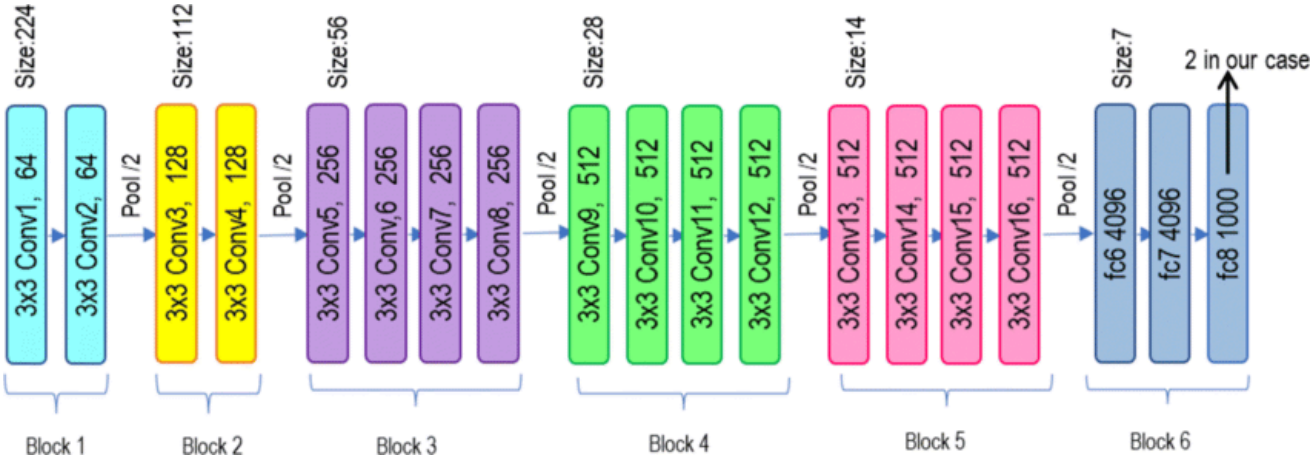
1. ViT



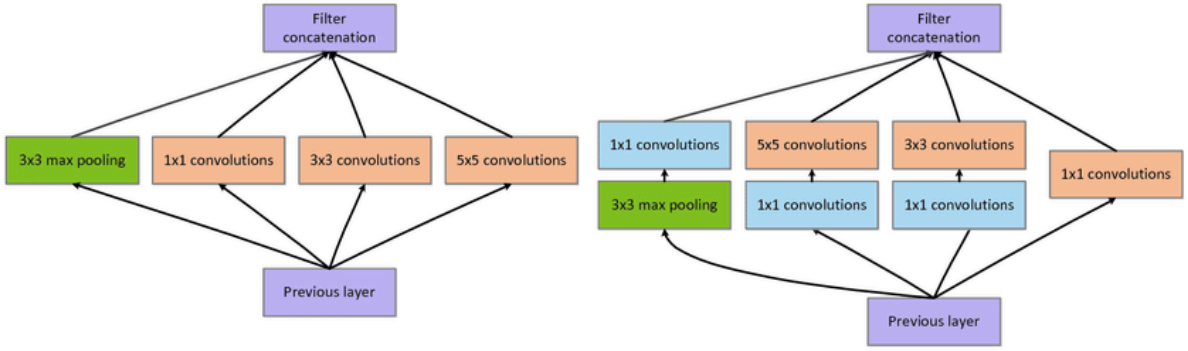
2. ResNet



3. **VGG19**



4. **Inception V1**



(a) Inception module, initial form

(b) Inception module with dimension reductions

3- References *(Respectively)*

1. ***AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE*** by Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby equal technical contribution, equal advising Google Research, Brain Team.
2. ***Deep Residual Learning for Image Recognition*** by Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Microsoft Research.
3. ***VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION*** by Karen Simonyan & Andrew Zisserman.
4. ***GOING DEEPER WITH CONVOLUTIONS*** by Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich.

4- Pros vs Cons

Architecture	Observed Behavior	Pros	Cons	Explanation
VGG19	Underfitting	• Very simple and well-understood architecture. • Strong ability to detect local edges and textures. • High theoretical capacity due to a large number of parameters.	• Extremely large number of parameters (inefficient). • No residual (skip) connections. • Suffers from vanishing gradient problem. • Difficult to optimize when trained from scratch.	Despite its large size, VGG19 underfits due to optimization failure rather than lack of capacity. Gradients diminish as they propagate backward, causing early layers to stop learning. As a result, the network fails to extract meaningful gesture features.
ResNet18	Overfitting	• Residual connections allow stable gradient flow. • Easy and fast to train. • Efficient and lightweight compared to deeper CNNs. • Strong hierarchical feature learning.	• Biased toward local texture rather than shape. • Prone to learning background or lighting shortcuts. • High risk of overfitting on small datasets.	ResNet18 learns too easily and memorizes training samples. Even with dropout and augmentation, it focuses on background textures or lighting patterns instead of hand geometry, leading to poor generalization.

Inception V1	Overfitting	• Multi-scale feature extraction (1×1, 3×3, 5×5 filters). • Computationally efficient. • Captures gestures at different sizes and distances.	• Complex architecture. • Sensitive to hyperparameters and initialization. • Still biased toward texture-based features.	Although parameter-efficient, Inception V1 is a strong learner. With limited gesture diversity, it overfits by memorizing recurring visual patterns instead of learning generalized hand shapes.
ViT	Good Generalization	• Uses self-attention to model global relationships. • Focuses on shape and geometry rather than texture. • Robust to background noise and occlusion. • Strong performance with pretraining.	• Typically data-hungry when trained from scratch. • Higher memory usage during inference. • Slower for high-resolution images.	ViT performs best because its attention mechanism captures global hand structure and finger relationships. It ignores background shortcuts that mislead CNNs, resulting in better validation performance and generalization.

5- Explanation of why ViT performs best for this task

Based on the evaluation, the ViT architecture has shown the best performance for the *Gesture Recognition* task, compared to **VGG19**, **ResNet18**, and **Inception V1**.

This outcome is due to how ViT processes visual information and how well its architectural bias aligns with the characteristics of the gesture dataset.

Gesture Recognition relies heavily on understanding the **global geometric structure** of the hand, rather than **fine-grained texture** details like skin patterns or lighting. ViT uses a self-attention mechanism that models relationships between all image regions simultaneously, unlike CNNs, which process images locally and are biased toward texture-based features.

In this project, **CNN-based models** *(ResNet18 and Inception V1)* have **overfitted** despite the use of data augmentation and regularization. These models learned shortcut features

such as background patterns, lighting conditions, or consistent textures present in the training data, leading to poor generalization on validation data.

VGG19, on the other hand, **underfitted** due to vanishing gradient, preventing it from learning gesture representations.

ViT outperformed the other architectures because it focuses on **global context rather than local texture**. Through self-attention, ViT learns how different parts of the hand relate to one another, enabling it to isolate the hand structure from background noise. Additionally, the use of **pre-trained weights** allowed ViT to overcome the data limitations commonly present in gesture datasets, resulting in stable training and improved generalization.