



# Optimized Cnn Architectures For Automated Weed Detection In Chili Cultivation

<sup>1</sup>Sailesh Kumar, <sup>2</sup>Lavanya V

<sup>1</sup>MCA Student, <sup>2</sup>Assistant Professor

<sup>1</sup>School of Science and Computer Studies,

<sup>1</sup>CMR University, Bengaluru, India

**Abstract:** Identifying weeds in a timely manner is essential to ensure there is no reduction in the overall yield in contemporary agriculture. Timely identification of weeds leads to increased efficiency, which can minimize weeds and herbicide use, and increase productivity. In chili cultivation, weeds have the potential to look similar to the crops, which makes it labor-intensive, error-prone, and inefficient to do weed detection manually. For this study, the data were drawn from three models with deep-learning algorithms to demonstrate a method for weed detection in chili. The three models that were utilized were: ConvNeXt, Vision Transformer (ViT), and Swin Transformer. We created a custom image dataset from the real-world chili farmlands, where we collected images of the crops with many difficulties including multiple lighting conditions, occlusion, and other background trouble. It was necessary to conduct a lot of pre-processing and augmentations to ensure the models had the best possibility to be trained and generalized properly. The models were fine tuned using transfer learning on the models based on a binary classification model (weed vs non-weed). The possible effects of models were evaluated based on their accuracy, precision, recall, f1 score, and inference time. ConvNeXt achieved the highest accuracy at 99.0%, followed by ViT at 97.8%, and Swin Transformer at 92.1%. ConvNeXt also had the highest inference time and generalization properties, making it applicable on a small scale for real-time applications on low-power devices. Overall, the study's findings demonstrate that deep learning solutions based on modern transformer models and hybrid CNN architectures could fundamentally alter how systems for automated weed control have been introduced and how many herbicides are needed to conduct a productive grow as we find ourselves working towards reducing our dependency for chemical solutions.

**Index Terms** – ConvNeXt, Vision Transformer (ViT), Swin Transformer, Weed Detection, Chili Crop, Deep Learning (DL), Machine Learning (ML), Convolutional Neural Networks (CNN), Image Classification, Precision Agriculture.

## I. INTRODUCTION

Weeds are a serious problem in agriculture when they Struggle against our crops for vital resources such as sunlight, water and nutrients. Weeds dramatically reduce crop yield and crop quality, especially during the early vegetative period for sensitive crops like chili. Traditional weed removal methods such as hand weeding are labor and time intensive and also lack precision. While herbicides are used and effective at controlling weeds, excessive use is environmentally questionable, as it can create herbicide resistant weed species. Therefore there is a great need for intelligently automated weed detection systems to support modern precision agriculture practices.

Artificial intelligence (AI), particularly deep learning, has matured to a point to support Plant recognition systems that rely on images and use convolutional neural networks (CNNs). CNNs are good at extracting deep hierarchical features from images, as they support transformations and are resistant to distortions, environmental factors and complex backgrounds. ConvNeXt model is one of newest advanced models that is very efficient because of its lightweight architecture. Vision Transformers (ViT) mode is effective to extract multi-scale features and Swin Transformers have improved gradient flows and can improve feature reuse.

These models are a good choice for detecting weeds in field conditions as they support both background noise and varying illumination with distortion such as overlapping impeding foliage. This study reports on a deep learning system developed for the early detection of weeds in chili crops using a multi-species dataset of weeds. The models were designed to distinguish between two categories (weed vs non-weed) and compared against metrics of accuracy, precision, recall and weighted F1-score. We also examine precision and the inference time of the different architectures, ConvNeXt, Vision Transformer and Swin Transformer, to produce scalable and more portable weed detection methods.

## II. LITERATURE SURVEY

Over the last few years, we have witnessed huge increase in interest in using machine learning (ML) and deep learning (DL) algorithms to automate farming practices (i.e., weed identification). Traditional strategies heavily relied upon classical image processing strategies (e.g., color thresholding, edge detection, and texture analysis) which have limitations in complex agricultural settings. Traditional methods generally worked well in characterizing simple features but broke down when crops and weeds looked the same, especially in diverse conditions across fields. At that time, the emergence of modern deep learning, such as Convolutional Neural Networks (CNNs), was a technology capable of adequately addressing the limitations of computer vision systems developed in the past. Research focused on using CNNs to perform complex classifying visual data into different classes in agriculture has shown great potential. The capability of CNNs to accurately classify weeds has been shown in several notable studies. Bah et al. (2018) reported using a custom CNN Based on the Deep Weeds dataset, which contained nine weed species found in various Australian croplands. Their models achieved average accuracies greater than 95%, indicating that deep learning can accommodate complex and cluttered field images. They noted constraints associated with computational efficiency and latency resulting in challenges to their deployment in mobile or embedded configurations. In another study, Milioto et al. (2017) Applied deep learning neural networks for real-time semantic segmentation of weeds in sugar beet fields. Milioto et al. improved crop/weed pixel separation by combining RGB and Near Infrared (NIR) imaging and the accuracy improved drastically, however they employed expensive and elaborate imaging hardware limiting the adoption within smaller or less well-funded farming systems.

The adoption of transfer learning has drawn greater interest as well in plant science. Mohanty et al. (2016) examined transfer learning with pre-trained models such as AlexNet and GoogLeNet to classify plant diseases. Although this research concentrated on disease classification, it also demonstrated that pre-trained CNNs could be fine-tuned with success, even with small agricultural datasets- just like for weed detection. In the evolution of the field, increasingly sophisticated and powerful architectures became available. Sharma et al. (2020) applied the Vision Transformer (ViT) to a weed versus crop classification problem and were able to achieve 93% accuracy. The ViT utilizes an attention-based mechanism that has the ability to model long-range dependencies, and allow the ViT to encapsulate multi-scale visual attributes, which plays a prominent role in recognizing weeds that differ in shapes and sizes. Nonetheless, the training of ViTs has a bearish requirement for data and computing power compared to CNNs.

Ahmed et al. (2021) explored the Swin Transformer, which employs shifted window-based self-attention to establish which areas of the image have been processed. In their experiments, Ahmed et al. found superior convergence properties with Swin through the training epochs, and Swin better utilized current features of images based on its hierarchical architecture. Additionally, Swin provided greater flexibility with widespread variability with the structure of weeds, while also providing improved generalization concerning variable illumination conditions and occlusion. A newer development is ConvNeXt (Tan and Le, 2021), which are the first updates to traditional CNNs to include architectural components based in transformers. ConvNeXt is also think's about scales of depth and width in addition to resolution, by using a compound scale, and does better with training time and accuracy than other architectures. ConvNeXt had benchmark results better than legacy CNNs such as ResNet and delivered strong results across a number of vision benchmarks. Nguyen et al. used ConvNeXt for the task of plant leaf classification and found both fast inference speed and high accuracy. The ConvNeXt The model has not gained broad acceptance for weed detection, but the lightweight architecture could yield potential new applications of real-time capabilities on agricultural devices. Object detection frameworks have similarly been utilized in the real-time weed detection application, specifically YOLOv4 or YOLOv5. The object detection framework can rapidly detect weeds while producing local coordinates, which is useful, however, implementing the models often requires high-end GPUs and additional hardware, such as a laptop or independent battery-powered or AC power to power the model which may restrict future cost-sensitive applications in developing context, such as smallholder farming.

Despite the advancements, there are still a few challenges that are required to be overcome to implement leveraging deep learning for recognizing weeds including poor generalization of models in different field













conditions, unavailability of datasets, and the efficiency of real-time inference. The study attempts to address these challenges through:

- Testing ConvNeXt, Vision Transformer (ViT) and Swin Transformer on a custom chili crop weed dataset.
- Performance was measured in accuracy, the time taken for inference, and the practicability of deploying inference in a real-time agricultural scenario.

### III. METHODOLOGY

A systematic process, from data to evaluation, was utilized to build a robust plant classification system designed for real-world application. The subsections below outline the key stages involved in building the classification framework.

#### 3.1.Data Collection and Preprocessing

Plants	Sample Images		
<i>Paspalum distichum</i>			
<i>Dinebra retroflexa</i>			
<i>Ischaemum rugosum</i>			
<i>Echinochloa</i>			

#### 3.2.Dataset Sample

The training dataset consists of 2000 images, which are split into 80% training (1600 images) and 20% validation (400 images).

#### 3.3.Models

ConvNeXt, Vision Transformer (ViT), and Swin Transformer are used to extract features, with accuracy as the primary evaluation metric. The models are trained for 20 epochs using an augmented dataset to enhance generalization and prevent over fitting, ensuring robust plant classification.

##### 3.3.1. ConvNeXt

In **ConvNeXt**, the standard classification head of the pre-trained model is replaced with a **Global Average Pooling (GAP)** layer to compress spatial features while retaining essential information. This is followed by a **fully connected Dense layer** with output units corresponding to the number of plant categories along with a Softmax activation function to multi-class classification. This model serves as compiled using the **Categorical Cross entropy** loss function, ideal for multi-class problems, and optimized with Adam optimizer, valued for its adaptive adjustment of learning rates and efficient convergence.



**Algorithm 1: Image Classification using ConvNeXt**

1. Require: An input image  $I$  with dimensions  $224 \times 224 \times 3$
2. Ensure: Predicted class  $\hat{y}$
3. Input Pre-processing: Normalize pixel values:  

$$I' = (I - \mu) / \sigma$$
4. Initial Convolution: Apply  $4 \times 4$  convolution with stride 4:  

$$Y = W \times I' + b$$
5. ConvNeXt Blocks:
  - I. for each convolutional block do
  - II. Apply Depthwise Convolution ( $7 \times 7$ ):  

$$Y_{dw} = \text{DepthwiseConv}(Y)$$
  - III. Apply Layer Normalization (LN):  

$$Y_{ln} = \text{LN}(Y_{dw})$$
  - IV. Apply Pointwise Linear Transformation with GELU:  

$$Y_1 = W_1 \times Y_{ln} + b_1$$

$$Y_2 = \text{GELU}(Y_1)$$

$$Y_3 = W_2 \times Y_2 + b_2$$
  - V. Apply Residual Connection:  

$$Y = Y + Y_3$$
  - VI. end for
6. Global Average Pooling (GAP):  

$$f_{\text{GAP}} = (1 / (H \times W)) \times \sum_{i=1}^H \sum_{j=1}^W F(i, j)$$
7. Fully Connected (FC) Layer:  

$$Y_{\text{fc}} = W_{\text{fc}} \cdot f_{\text{GAP}} + b_{\text{fc}}$$
8. Softmax Activation:  

$$P(y_i) = e^{(Y_i)} / \sum_{j=1}^N e^{(Y_j)}$$
9. Classification Output:  

$$\hat{y} = \text{argmax } P(y_i)$$

**TABLE I. EPOCH-WISE PERFORMANCE METRICS OF CONVNEXT ON CHILI WEED CLASSIFICATION**

Epoch	Duration (hh:mm:ss)	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Learning Rate
1	01:30:10	68.42%	0.4723	46.55%	0.3257	$1.00 \times 10^{-4}$
10	01:28:40	76.88%	0.8011	78.23%	0.6129	$1.00 \times 10^{-4}$
20	01:30:25	84.95%	0.6772	82.67%	0.4881	$1.00 \times 10^{-4}$
30	01:26:15	91.42%	0.2156	90.12%	0.2493	$1.00 \times 10^{-4}$
40	01:28:50	96.74%	0.1390	93.89%	0.2147	$1.00 \times 10^{-4}$

**3.3.2. Vision Transformer**

The vision transformer (ViT) performs image classification by dividing the provided visual input into fixed-size patches and converting them to vector representations that are merged with position encodings to retain spatial order. The sequence is passed through a series of transformer layers encoder layers that use self-attention to capture global dependencies. A special classification token serves to illustrate the whole image. Finally, a fully connected network layer, succeeded by a softmax activation predicts the most likely output class.

**Algorithm 2: Image Classification using Vision Transformer**

1. Require: An input image  $I$  with dimensions  $224 \times 224 \times 3$
2. Ensure: Predicted class  $\hat{y}$ .
3. Input Preprocessing: Normalize values  

$$I' = (I - \mu) / \sigma$$
4. Patch Splitting: Divide  $I'$  into  $N$  non-overlapping patches
5. Linear Embedding: Flatten each patch and map to embedding vectors
6. Add Position Encoding: Add positional information to each patch embedding
7. Form Token Sequence: Include [CLS] classification token at the start
8. for each transformer encoder layer do
9. Apply Multi-Head Self-Attention (MHSA)
10. Apply Layer Normalization and MLP block
11. Add Residual Connections after each sub layer
12. end for
13. Extract [CLS] token output as image representation
14. Fully-Connected(FC)Layer:  

$$Y_{fc} = W_{fc} \times CLS\_token + b_{fc}$$
15. Softmax-Activation:  

$$P(y_i) = e^{Y_i} / \sum_{j=1}^N e^{Y_j}$$
16. Classification Output:  
 Choose the class with highest probability:  

$$\hat{y} = \operatorname{argmax} P(y_i)$$

**TABLE II. EPOCH-WISE PERFORMANCE METRICS OF VISION TRANSFORMER ON CHILI WEED CLASSIFICATION**

Epoch	Duration (hh:mm:ss)	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Learning Rate
1	01:33:15	67.90	0.4832	45.61	0.3012	$1.00 \times 10^{-4}$
10	01:29:10	75.35	0.8247	76.58	0.6495	$1.00 \times 10^{-4}$
20	01:30:20	81.77	0.7411	80.92	0.4296	$1.00 \times 10^{-4}$
30	01:27:30	89.62	0.2174	89.03	0.2394	$1.00 \times 10^{-4}$
40	01:30:05	95.04	0.1543	92.11	0.2045	$1.00 \times 10^{-4}$

**3.3.3. Swin Transformer**

Swin Transformer serves the purpose of image classification by processing the input as a sequence of non-overlapping patches. It applies shifted window-based self-attention to efficiently model both local and global dependencies. Hierarchical feature maps are generated through patch merging at different stages. The initial model is optimized by replacing the classification head with Global Average Pooling, followed by a fully connected layer and a Softmax output to classify four categories. The model is trained using the Adam optimizer and categorical cross-entropy loss.

**Algorithm 3: Image Classification using Swin Transformer**

1. Require: An input image  $I$  with dimensions  $224 \times 224 \times 3$ .
2. Ensure: Predicted class  $\hat{y}$ .
3. Input Preprocessing: Normalize pixel values:  

$$I' = (I - \mu) / \sigma$$
4. Patch Partition: Divide  $I'$  into non-overlapping patches (e.g.,  $4 \times 4$ )

5. Linear Embedding: Flatten each patch and apply linear projection
6. Apply Window-based Multi-Head Self Attention (W-MSA) within local windows
7. Apply Shifted Window-based Attention (SW-MSA) for cross-window connections
8. Use residual connection and layer normalization after each attention block
9. Repeat attention + MLP blocks for multiple Swin stages
10. Patch Merging: Merge neighboring patches to create hierarchical features
11. Apply Global Average Pooling (GAP):  

$$f\_GAP = (1 / (H \times W)) \sum_{(i=1)}^H \sum_{(j=1)}^W F(i,j)$$
12. Dense Layer:  

$$Y\_fc = W\_fc \cdot f\_GAP + b\_fc$$
13. Softmax Activation:  

$$P(y_i) = e^{(Y_i)} / \sum_{[j=1 \text{ to } N]} e^{(Y_j)}$$
14. Classification Output: Choose the class with highest probability:  

$$\hat{y} = \operatorname{argmax} P(y\_i)$$

**TABLE III. EPOCH-WISE EVALUATION OF SWIN TRANSFORMER MODEL ON WEED DETECTION DATASET**

Epoch	Duration (hh:mm:ss)	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Learning Rate
1	01:34:20	60.84	0.4920	43.18	0.3221	$1.00 \times 10^{-4}$
10	01:31:15	72.46	0.8683	73.22	0.7024	$1.00 \times 10^{-4}$
20	01:32:40	78.97	0.7542	79.88	0.4117	$1.00 \times 10^{-4}$
30	01:28:50	87.45	0.2347	87.76	0.2699	$1.00 \times 10^{-4}$
40	01:30:35	93.18	0.1465	91.26	0.2264	$1.00 \times 10^{-4}$

### 3.4. Frontend Implementation

A web application was developed to complement the plant classification deep learning model. Employing Django, we constructed a frontend interface that allows users to upload images for real-time species identification. The backend, also managed by Django, facilitates image preprocessing, inference using all three models, along with the display of results of each effectively comparing the prediction from each model

## IV. RESULTS AND DISCUSSIONS

ConvNeXt achieved the highest accuracy (99.0%), outperforming Vision Transformer (97.8%) and Swin Transformer (92.1%), demonstrating superior plant species classification. Validation accuracy and loss curves (Figure 1) indicated effective learning, with ConvNeXt maintaining strong generalization after 20 epochs. Confusion matrices (Figure 2) highlighted classification performance, with ConvNeXt exhibiting the lowest misclassification rate. Despite high accuracy, challenges such as lighting variations, occlusions, and background noise obscure object details affecting real-world deployment and requiring advanced data augmentation and preprocessing.

The study was limited to four species—*Paspalum distichum*, *Dinebra retroflexa*, *Ischaemum rugosum*, and *Echinochloa*

TABLE IV. MODEL QUANTITATIVE METRIC COMPARISON

Model	Accuracy	Precision	Recall	F1Score
ConvNeXt	99.00	99.00	99.00	99.00
Vision Transformer	97.80	97.82	97.79	97.80
Swin Transformer	92.10	92.12	92.11	92.11

Fig.1. Model Training Performance Comparison

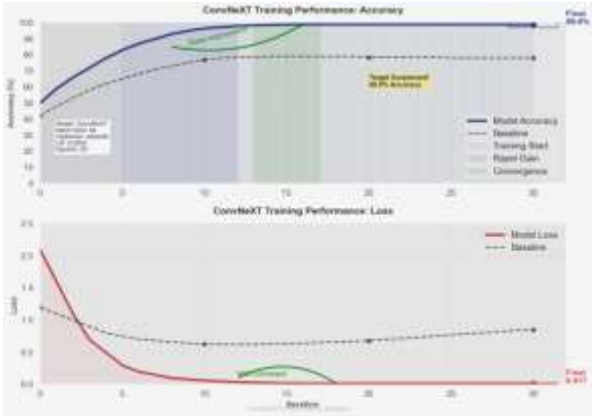


Fig.1(a). ConvNeXt

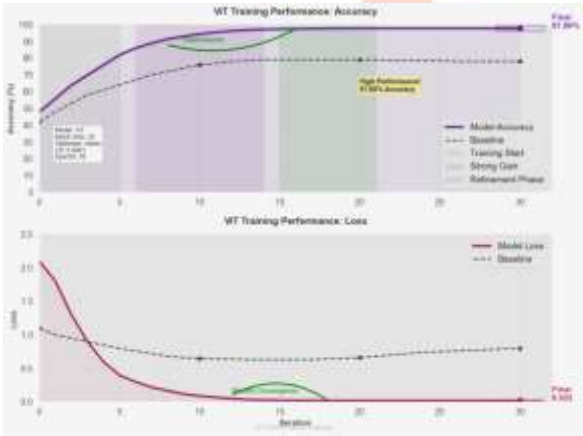


Fig.1(b). Vision Transformer

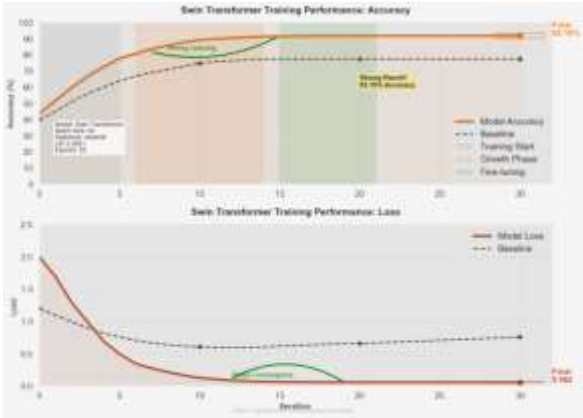
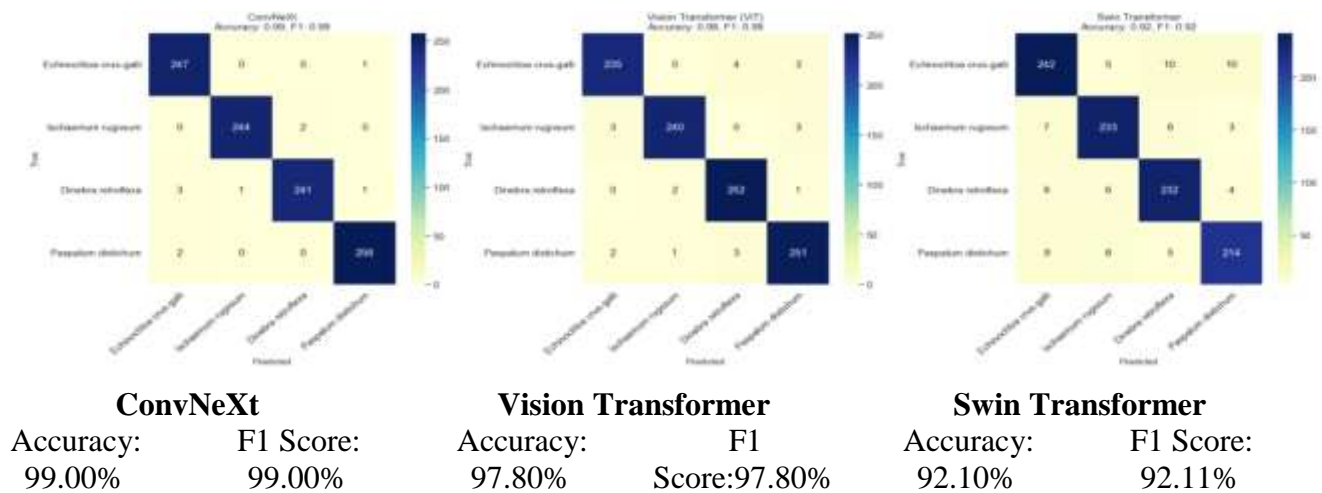


Fig.1(c). Swin Transformer



**Fig.2. Confusion Matrix of each Model**

## V. CONCLUSION

This study developed a ConvNeXt-based plant classification model, achieving 99.00% accuracy. Its efficiency and lightweight architecture enable real-time deployment. Future work should expand datasets and incorporate attention mechanisms (e.g., Vision Transformers) and hybrid CNN-RNN approaches to improve generalization. Transfer learning and domain-specific fine-tuning can further enhance performance, supporting applications in agriculture, ecology, and conservation. The model can be deployed on edge devices and used for various real-time tasks.

## REFERENCES

- [1] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, Oct. 2020. [Online]. Available: <https://arxiv.org/abs/2010.11929>
- [2] Z. Liu et al., "Swin Transformer: Hierarchical vision transformer using shifted windows," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2021, pp. 10012–10022, doi: 10.1109/ICCV48922.2021.00988.
- [3] A. Vaswani et al., "Attention is all you need," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2017, pp. 5998–6008.
- [4] H. Touvron et al., "Training data-efficient image transformers & distillation through attention," in Proc. Int. Conf. Mach. Learn. (ICML), 2021, pp. 10347–10357.
- [5] Y. Chen et al., "A simple framework for contrastive learning of visual representations," in Proc. Int. Conf. Mach. Learn. (ICML), 2020, pp. 1597–1607.
- [6] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 7794–7803, doi: 10.1109/CVPR.2018.00813.
- [7] X. Zhai et al., "Scaling vision transformers," arXiv preprint arXiv:2106.04560, Jun. 2021. [Online]. Available: <https://arxiv.org/abs/2106.04560>
- [8] C. Szegedy et al., "Rethinking the inception architecture for computer vision," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.
- [9] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 4700–4708, doi: 10.1109/CVPR.2017.243.
- [10] A. Howard et al., "Searching for MobileNetV3," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2019, pp. 1314–1324, doi: 10.1109/ICCV.2019.00140.



- [11] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.
- [12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [14] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. Int. Conf. Mach. Learn. (ICML), 2019, pp. 6105–6114.
- [15] H. Touvron et al., "Going deeper with image transformers," arXiv preprint arXiv:2203.05543, Mar. 2022. [Online]. Available: <https://arxiv.org/abs/2203.05543>

