

# DEEP\_TRANSFER LEARNING

Ragu surya Prakash

# Efficient Net: A Pretrained Model for Image Classification

## Key Features:

- **Compound Scaling:** Efficient Net scales all dimensions of the network (width, depth, resolution) using a set of fixed scaling coefficients.
- **Model Variants:** EfficientNet-B0 to EfficientNet-B7, where each successive model offers increased accuracy and computational cost.
- **Pretrained Models:** Available for various tasks including image classification, object detection, and more.

## Benefits:

- **High Accuracy:** Achieves better accuracy on ImageNet compared to previous models like ResNet and Inception.
- **Efficiency:** Reduces the number of parameters and FLOPs, making it suitable for deployment in resource-constrained environments.
- **Transfer Learning:** Pretrained EfficientNet models can be fine-tuned for specific tasks, leveraging the learned features from large datasets.

- **Usage:**

TensorFlow and Keras: EfficientNet models are

integrated into TensorFlow and Keras, making them easy to use.

- **Example Code:**

- `from tensorflow.keras.applications import EfficientNetB0`
- `# Load the pretrained model`
- `model = EfficientNetB0(weights='imagenet')`
- `# Example usage`
- `from tensorflow.keras.preprocessing import image`
- `from tensorflow.keras.applications.efficientnet import preprocess_input, decode_predictions`
- `import numpy as np`
- `img_path = 'path_to_your_image.jpg'`
- `img = image.load_img(img_path, target_size=(224, 224))`
- `x = image.img_to_array(img)`
- `x = np.expand_dims(x, axis=0)`
- `x = preprocess_input(x)`
- `preds = model.predict(x)`
- `print ('Predicted:', decode_predictions(preds, top=3)[0])`

- **Resources and Links:**

**Google Research Blog:** EfficientNet:

Rethinking Model Scaling for Convolutional Neural Networks

- <https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html>
- **TensorFlow Documentation:** EfficientNet in TensorFlow
- [https://www.tensorflow.org/api\\_docs/python/tf/keras/applications/EfficientNetB0](https://www.tensorflow.org/api_docs/python/tf/keras/applications/EfficientNetB0)
- **GitHub Repository:** TensorFlow Models
- <https://github.com/tensorflow/models>

<https://www.semanticscholar.org/paper/EfficientNet%3A-Rethinking-Model-Scaling-for-Neural-Networks/4f2eda8077dc7a69bb2b4e0a1a086cf054adb3f9>

