Implementing Efficient Vision Transformers

Guhan S

42733024

B.E. CSE DATA SCIENCE

Sathayabama Institute Of Science And Technology

Paper Summary:

This paper explores the efficiency of Vision Transformers (ViTs) by evaluating their computational cost, memory usage, accuracy, robustness, and fairness. It introduces the Efficiency-360 framework, which assesses ViTs based on multiple factors such as privacy, transparency, and adversarial robustness. The study compares various ViT architectures, including DeiT, Swin Transformer, PVT, and others, focusing on techniques that enhance efficiency, such as hierarchical feature extraction, token mixing, and knowledge distillation.

The research highlights that DeiT improves training efficiency through knowledge distillation, while Swin Transformer uses hierarchical window-based attention to reduce computational complexity for dense vision tasks like object detection and segmentation. The study concludes that hybrid models combining convolutional and transformer-based approaches provide a balance between performance and efficiency. Future research should focus on reducing computational overhead, improving fairness in AI models, and making ViTs more adaptable to real-world applications.

Implementation:

To implement Vision Transformers efficiently, it is essential to install the necessary libraries that support deep learning, image processing, and model visualization. The required libraries include Torch and Torchvision for deep learning operations, Timm for pre-trained Vision Transformer models, Streamlit for building an interactive user interface, Pillow and NumPy for image processing, and Matplotlib and Grad-CAM for model interpretability and visualization.

1.Define Available Vision Transformers

- Only two models are included: DeiT and Swin Transformer (since they are efficient and widely used).

- Each model has specific parameters, FLOPs (computational cost), and Top-1 accuracy.
- The model details are stored in a dictionary.

2.Create a Streamlit UI for Model Selection

- The user interface (UI) is built using Streamlit.
- A dropdown menu allows users to choose between DeiT and Swin Transformer.
- The UI displays model performance metrics, including:
    - Number of parameters (M)
    - FLOPs (Giga FLOPs)
    - Top-1 accuracy on ImageNet

3.Load the Selected Model

- The selected model (DeiT or Swin Transformer) is loaded using timm.
- The model is pre-trained and set to evaluation mode to prevent unnecessary gradient calculations.
- Streamlit caching is used to prevent reloading the model every time.

4.Define Image Preprocessing

- The input image is resized to 224x224 pixels (required by ViTs).
- The image is converted to a tensor and normalized using ImageNet mean and standard deviation.
- The preprocessing ensures compatibility with pre-trained models.

5.Upload an Image for Classification

- Users can upload an image in .jpg, .png, or .jpeg format.
- The uploaded image is displayed in the Streamlit UI.
- It is preprocessed before feeding it into the model.

6.Perform Inference (Image Classification)

- The preprocessed image is passed through the model in no-grad mode (faster inference).

- The output class probabilities are computed, and the highest-scoring class is selected.
- The class label is retrieved from ImageNet using an online JSON file.

7.Display Prediction Results

- The predicted class label and ID are displayed.
- The inference time (speed of model prediction) is also shown.
- This helps users compare the computational efficiency of DeiT vs. Swin Transformer.

8.Select the Correct Target Layer for Grad-CAM

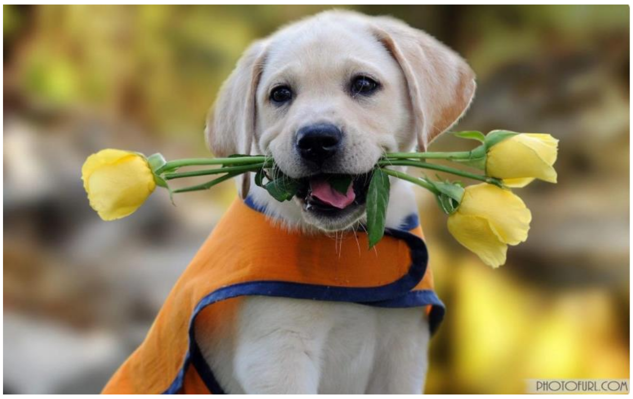
- EigenCAM is used to visualize model attention on the input image.
- Different transformers use different final attention layers:
  - DeiT → model.blocks[-1].norm1
  - Swin Transformer → model.layers[-1].blocks[-1].norm1
- The correct layer is automatically selected based on the chosen model.

9.Compute and Display EigenCAM Heatmap

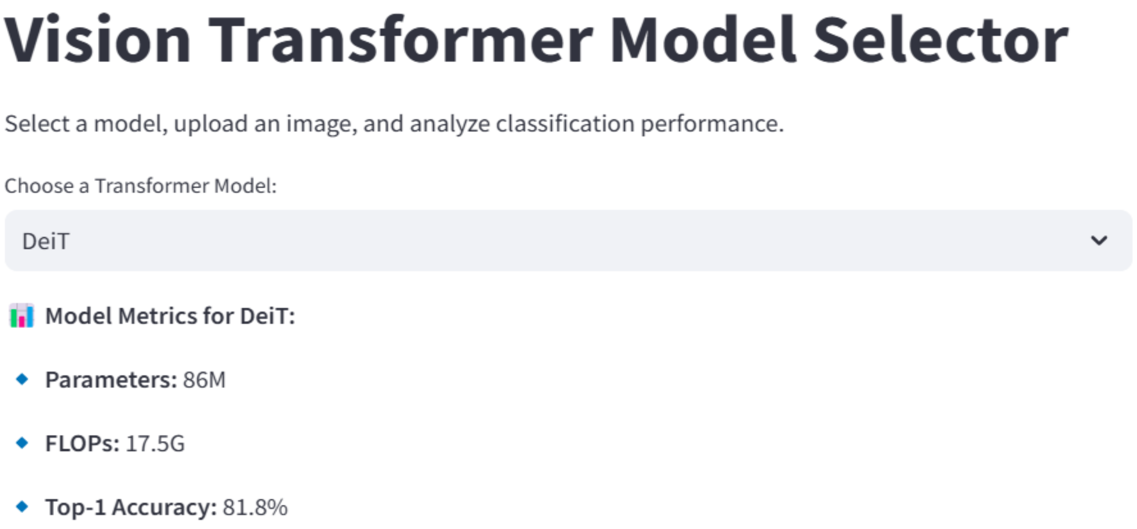
- The EigenCAM heatmap is computed on the selected model's attention layer.
- The heatmap is resized to match the original image size (fixes shape mismatch issues).
- The overlayed heatmap is displayed in the UI to highlight model attention.

Observation And Results:

Sample Data



DeiT Metrics:



Vision Transformer Model Selector

Select a model, upload an image, and analyze classification performance.

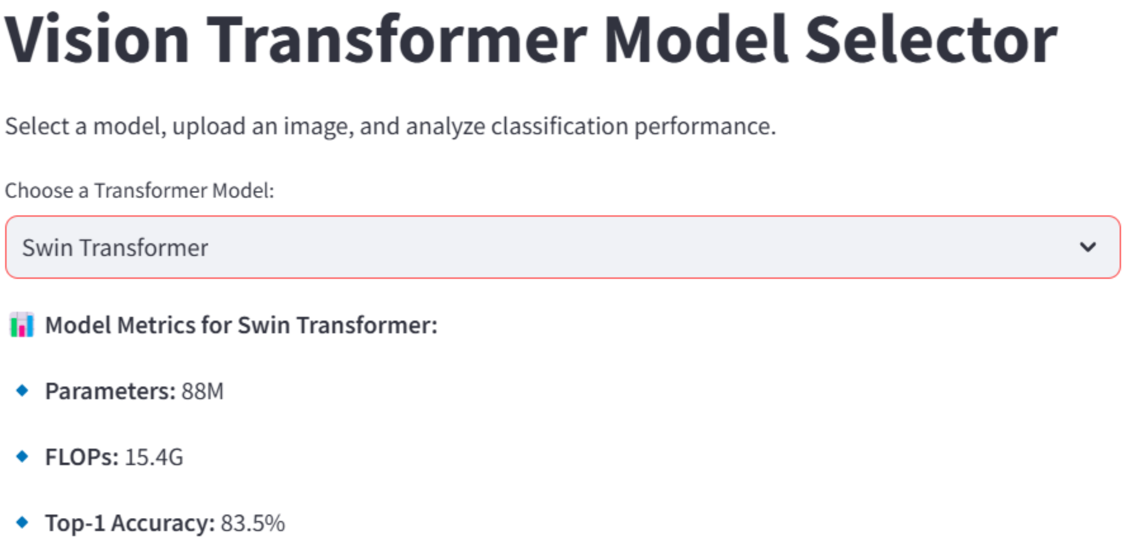
Choose a Transformer Model:

DeiT

📊 **Model Metrics for DeiT:**

- **Parameters:** 86M
- **FLOPs:** 17.5G
- **Top-1 Accuracy:** 81.8%

Swin Metrics:



Vision Transformer Model Selector

Select a model, upload an image, and analyze classification performance.

Choose a Transformer Model:

Swin Transformer

📊 **Model Metrics for Swin Transformer:**

- **Parameters:** 88M
- **FLOPs:** 15.4G
- **Top-1 Accuracy:** 83.5%

Class Prediction:

For Swin Model:

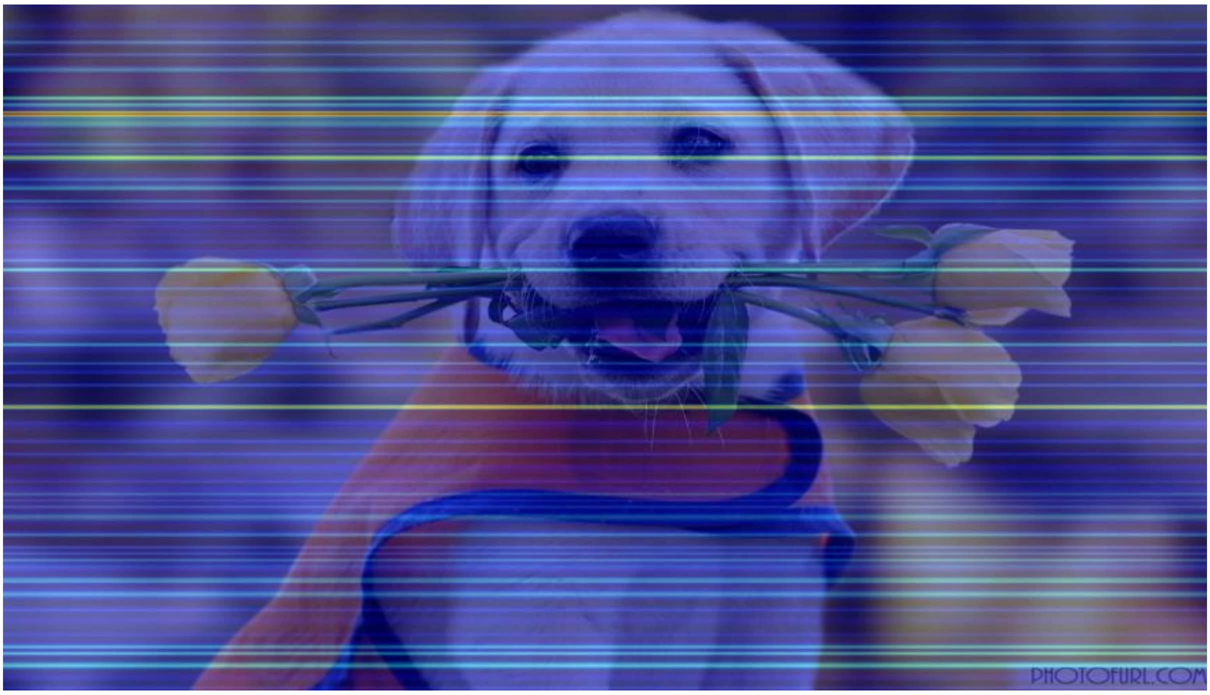
- Predicted Class: Labrador_retriever (ID: 208)
- Inference Time: 0.4707 seconds

For DeiT Model:

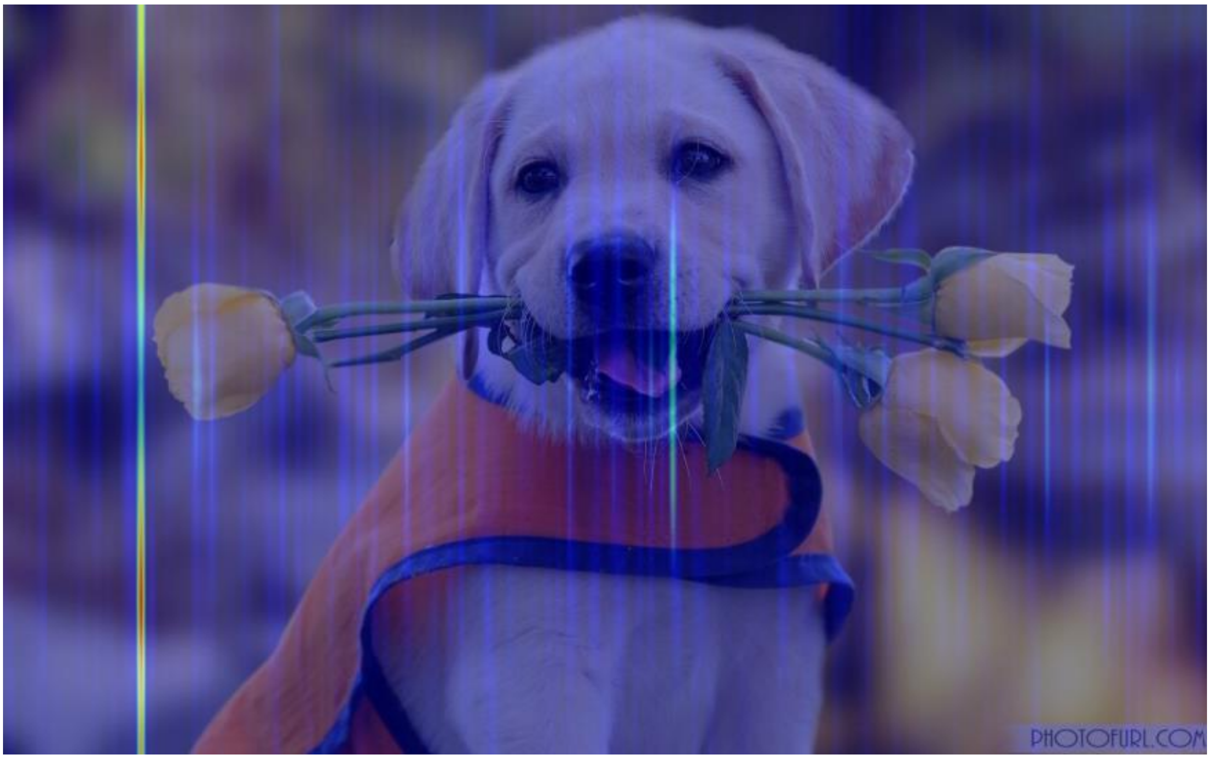
- Predicted Class: Labrador_retriever (ID: 208)
- Inference Time: 0.7295 seconds

Eigen CAM Visualization (Heatmap):

For DeiT:



For Swin:


PHOTOFURL.COM