INTER IIT TECH MEET 13.0

PATCH PRUNING IN VISION TRANSFORMERS

Team ID: 39

ENGINEERS' CONCLAVE

IDEATION

Vision Transformers (ViTs) excel in visual tasks by modeling global context via self-attention, but this leads to high computational and memory costs, especially when processing redundant image patches. The quadratic growth of MHSA computations with the number of patches exacerbates inefficiencies.

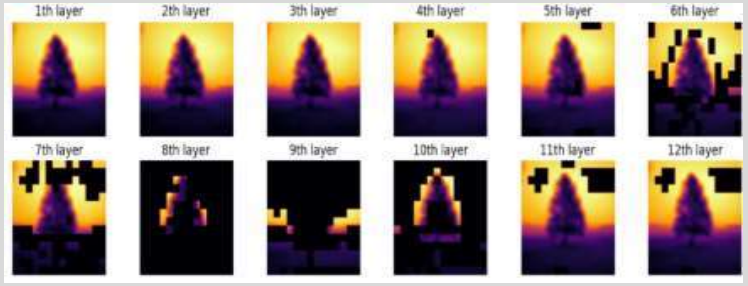
This work introduces a novel pruning approach for ViTs that balances computation and accuracy, while enhancing the model's explainability by identifying and skipping redundant patches.

PROTOTYPE DESCRIPTION

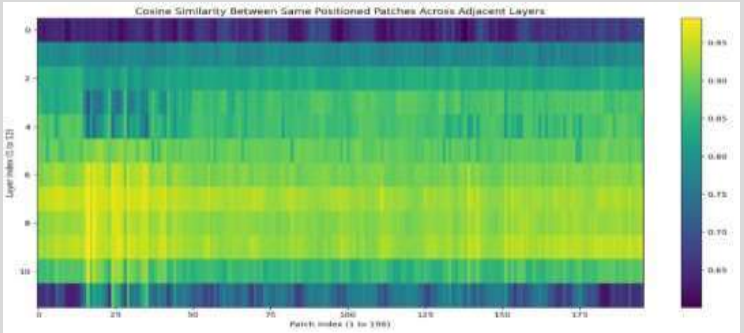
Our mode aims to optimize Vision Transformers by dynamically pruning redundant patches using cosine similarity. A lightweight MLP predicts pruning scores to balance computation and accuracy while ensuring adaptability and model explainability

The prototype employs an MLP to predict pruning scores based on cosine similarity between patches.

During training, pruning labels are generated and refined, while at inference, patches are skipped or retained based on these scores, ensuring efficient and adaptive performance.



Above image depicts the layer wise pruned patch outputs, for 'tree' labelled image from CIFAR 100



Following training and the testing workflows are employed in our approach

1.1 Training Workflow

Algorithm 1 Training Workflow

1: Compute scaled cosine similarity between corresponding patches across layers i and $(i+1)$
2: Generate labels based on cosine_threshold:
3: **if** scaled cosine similarity > cosine_threshold **then**
4: Label = 0 (Skip Patch)
5: **else**
6: Label = 1 (Keep Patch)
7: **end if**
8: Use a lightweight MLP to predict a pruning score between [0, 1] for each patch, which is inversely proportional to the scaled cosine similarity score
9: Compute the Cross-Entropy loss between the generated labels and the MLP-predicted scores
10: Backpropagate the loss to optimize the MLP
11: **Note:** Skipped patches can be reintroduced in subsequent layers

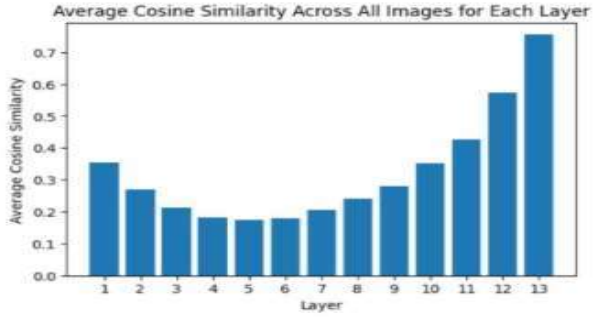
1.2 Inference Workflow

Algorithm 2 Inference Workflow

1: For each patch, feed it into the lightweight MLP to compute the pruning score
2: Apply threshold for binary decision:
3: **if** MLP score > score_threshold **then**
4: Keep Patch (Label = 1)
5: **else**
6: Skip Patch (Label = 0)
7: **end if**
8: Accuracy Calculation:
9: Compute the scaled cosine similarity for each patch
10: Generate ground truth labels based on the scaled cosine similarity and cosine_threshold
11: Compare the ground truth labels with the predicted labels from the MLP to compute accuracy

BASIS OF THE APPROACH

We observed that, the average cosine similarity between the patches withing a layer is first decreasing and then rising to large values such as 0.6, 0.7, Also, we can observe that, as we go deep into the layers, the patches at same location have very close similarity, approaching nearly 0.9. This redundancy can be resolved by selectively pruning similarly positioned patches across the layers.



RESULT & CONCLUSION

Our patch pruning approach reduces the computational complexity of Vision Transformers (ViTs) while preserving accuracy. By dynamically identifying redundant patches using cosine similarity, it enhances efficiency and offers insights into ViT behavior. This improves model explainability and promotes more interpretable, resource-efficient models.