VIETNAM NATIONAL UNIVERSITY OF HO CHI MINH CITY

UNIVERSITY OF INFORMATION TECHNOLOGY

FACULTY OF COMPUTER SCIENCE

PROJECT REPORT

SUBJECT: MULTIMEDIA INFORMATION RETRIEVAL

TOPIC: IMAGE RETRIEVAL

Class:	CS336.O11.KHCL
Instructor:	PhD. Ngo Duc Thanh
Students:	Nguyen Vu Duong - 20520465
	Le Tran Quoc Khanh - 20520574
	Huynh Minh Quan - 20520707

Ho Chi Minh City, January 3, 2024

CONTENTS

I. OVERVIEW

1. Introduction

In today's information age, images have become a dominant form of communication. From social media and e-commerce platforms to scientific research and cultural archives, visual data permeates every aspect of our lives. Navigating this ever-expanding sea of images requires efficient and intuitive retrieval systems.

Traditional keyword-based methods have long served as the cornerstone of image retrieval. By associating images with descriptive tags and captions, these systems allow users to search for images based on their semantic content. While useful for simple queries, such methods have significant limitations:

- Subjectivity and ambiguity: Keywords can be interpreted differently by users and indexers, leading to inconsistent results.
- Limited vocabulary: Not all image features can be easily described with words, particularly subtle details or abstract concepts.
- Manual annotation: Keyword tagging is a time-consuming and laborious process, often requiring expert knowledge.

These limitations often result in frustrating user experiences, with irrelevant images cluttering search results and relevant ones remaining hidden. As the volume and complexity of image data continue to grow, the need for a more sophisticated approach becomes increasingly evident.

Herein lies the motivation for building content-based image retrieval systems. By directly analyzing the visual content of images, these systems offer a powerful alternative to keyword-based methods. Leveraging the capabilities of deep learning techniques, they can extract high-dimensional features that capture the essence of an image, beyond the limitations of human language. This opens up exciting possibilities for:

- Accurate and intuitive image search: Users can search for images using examples or visual concepts, regardless of their ability to articulate precise keywords.
- Efficient annotation and organization: Automatically extracted features can be used to efficiently tag and organize large image collections, saving time and resources.
- Applications: Content-based retrieval unlocks possibilities in various fields like medical diagnosis, architectural design, and remote sensing, where visual similarity plays a crucial role.

Building a robust and effective content-based image retrieval system presents a significant challenge, but the potential rewards are immense. By delving into the field of deep learning and leveraging the power of visual analysis, we can unlock a new era of image discovery and navigation, where users can find what they need, visually, intuitively, and with ease.

2. Project Objectives

This project aims to develop a content-based image retrieval system using deep learning techniques. Our objective is to create a system that enables users to find visually similar images based on a query image or set of visual features.

Specifically, we focus on retrieving images from two datasets:

- The Paris Dataset
- The Oxford Buildings Dataset

To assess the effectiveness of our system, we will utilize the following performance metrics:

- Non-interpolated Mean Average Precision (Non-interpolated MAP): This metric measures the system's accuracy at specific retrieval thresholds, providing a detailed view of its performance across different levels of precision and recall.
- Interpolated Mean Average Precision (Interpolated MAP): This metric smooths out the precision-recall curve, providing a more holistic assessment of the system's overall retrieval performance.
- Retrieval time: This metric measures the system's efficiency in terms of the time it takes to process a query and return relevant results, ensuring a responsive user experience.

By evaluating these metrics for various deep learning models and configurations, we aim to establish the most effective approach for our image retrieval system, balancing accuracy with efficiency to deliver optimal visual search capabilities.

3. Dataset Insights

3.1. The Paris Dataset

The Paris Dataset consists of 6,412 images of Paris landmarks, collected from Flickr. Each image is labeled with the landmark it depicts, making it a valuable resource for tasks like landmark recognition, image retrieval, and scene understanding. The dataset is available for download under the terms of Flickr's Terms of Use and the VGG group's own Terms of Access.

The Paris Dataset

The images cover a wide range of Parisian landmarks, including the Eiffel Tower, the Louvre Museum, the Arc de Triomphe, and Notre Dame Cathedral. They are captured from a variety of viewpoints and under different lighting conditions, making the dataset challenging but realistic. In addition to the image itself, each data point also includes the landmark name, the camera viewpoint (e.g., front, back, side), and the date the image was taken.

The Paris Dataset has been used for a variety of research projects in computer vision, including landmark recognition, image retrieval, scene understanding, structure from motion.

This is a valuable resource for researchers and developers working on computer vision tasks related to landmark recognition, image retrieval, and scene understanding.

3.2. The Oxford Buildings Dataset (Oxford5k)

The Oxford Buildings Dataset, also known as Oxford5k, is a large-scale image dataset created by the Visual Geometry Group at the University of Oxford. It consists of 5,062 high-resolution images of 11 landmark buildings in Oxford, England. Each building has five different views captured from various distances and angles.

The dataset is widely used in computer vision research for tasks such as image classification, object detection, image retrieval, and structure from motion. It is known for its challenging nature due to the variations in viewpoint, illumination, and scale of the images.

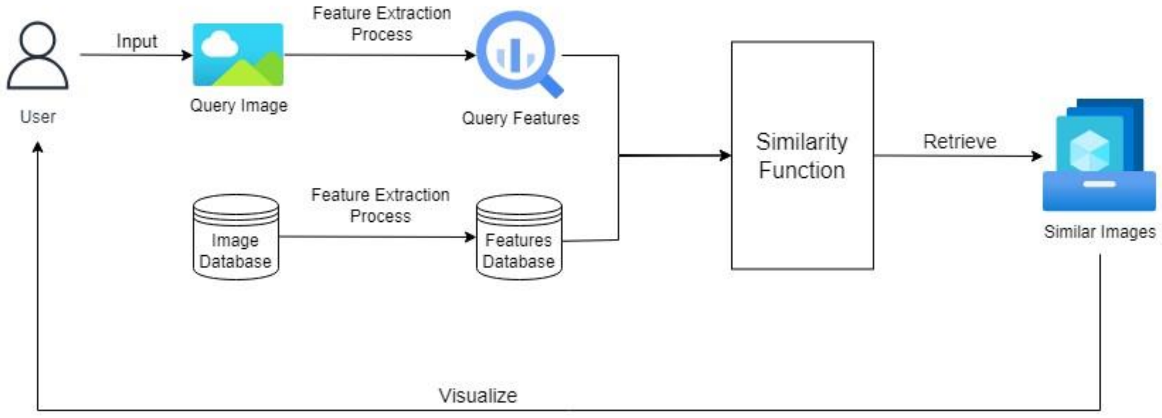
The Oxford Buildings Dataset (Oxford5k)

Oxford5k has been used in numerous research papers and has contributed to the development of many state-of-the-art computer vision algorithms. It is a valuable resource for researchers and developers working in the field of computer vision.

II. METHODOLOGY

1. Workflow

Our system take an image from user as query. Using Bag of Visual Words to extract features of input then calculate similarity between input's features and extracted features of images in datasets. The result is a list of images ranked by relevant in descending order. Here is the workflow of our system.



2. Feature Extraction

Image feature extraction is the process of identifying and extracting distinctive characteristics from an image that can be used to represent its content. These features are then used for various purposes, such as image classification, object detection, image retrieval, and image segmentation.

There are many different methods for image feature extraction, which can be broadly categorized into two main approaches:

- **Traditional Methods:** These methods are based on handcrafted algorithms and mathematical models that are designed to extract specific types of features. Some popular traditional methods: Color Histograms, Texture Descriptors, Shape Features, SIFT (Scale-Invariant Feature Transform) and SURF (Speeded Up Robust Features).
- **Deep Learning Methods:** Deep learning has revolutionized image feature extraction in recent years. Deep learning models, such as convolutional neural networks (CNNs) and transformers, can automatically learn complex and abstract features from images without the need for hand-crafted features. Deep learning have achieved state-of-the-art performance in many image processing tasks, including image classification, object detection, image segmentation and image retrieval.

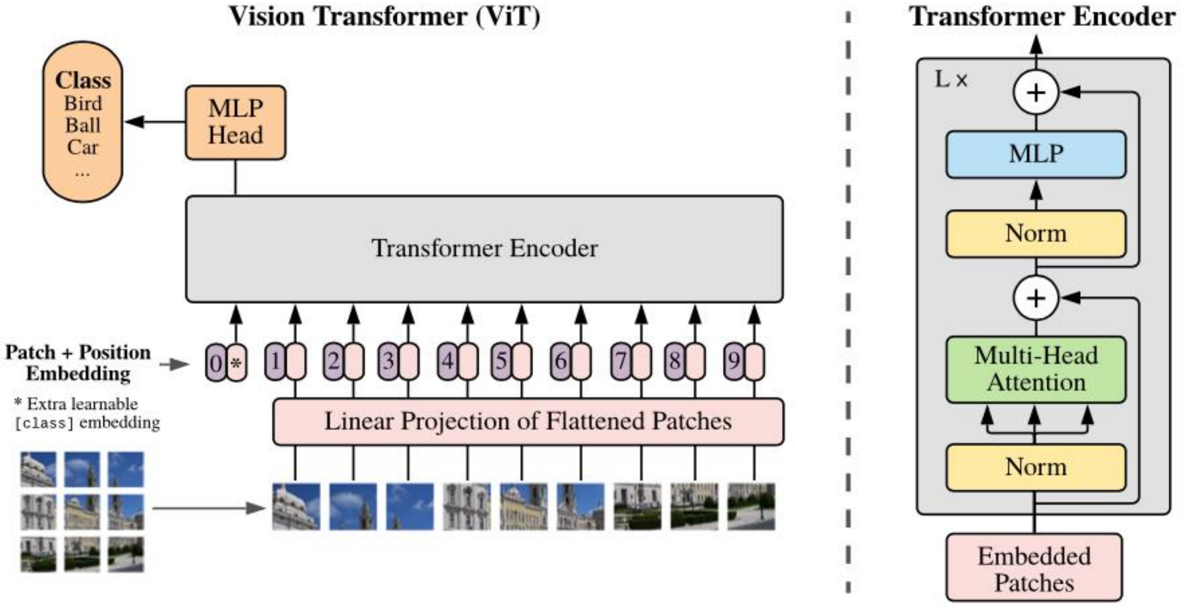
The choice of image feature extraction method depends on the specific task at hand. Traditional methods may be sufficient for simple tasks, such as image classification of basic objects. However, deep learning methods are often the best choice for more complex tasks, such as object detection in cluttered scenes or image segmentation of fine-grained details.

For this project, we leverage a combination of deep learning architectures. These choices are rooted in the project's objectives of achieving both high accuracy and reasonable efficiency in image retrieval.

2.1. VisionTransformer

Vision Transformers (ViT) is an architecture that uses self-attention mechanisms to process images. The Vision Transformer Architecture consists of a series of transformer blocks. Each transformer block consists of two sub-layers: a multi-head self-attention layer and a feed-forward layer.

The self-attention layer calculates attention weights for each pixel in the image based on its relationship with all other pixels, while the feed-forward layer applies a non-linear transformation to the output of the self-attention layer. The multi-head attention extends this mechanism by allowing the model to attend to different parts of the input sequence simultaneously.



Vision Transformer Architecture

ViT also includes an additional patch embedding layer, which divides the image into fixed-size patches and maps each patch to a high-dimensional vector representation. These patch embeddings are then fed into the transformer blocks for further processing.

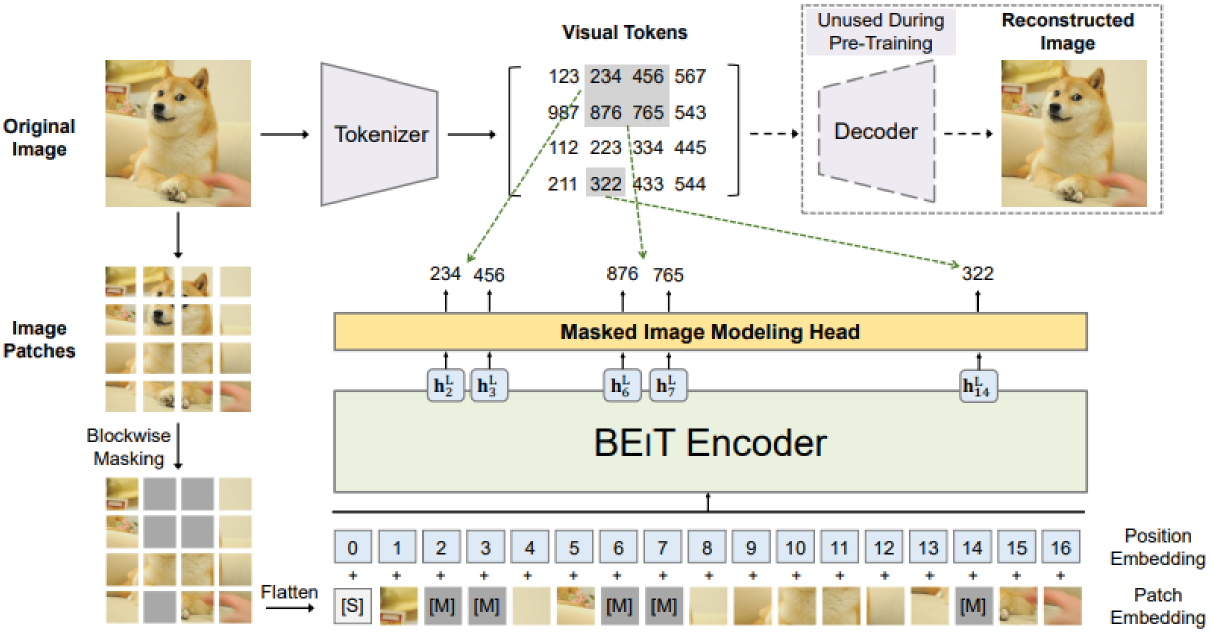
Overview of the model: Split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder.

Vision Transformers represent a significant leap forward in image recognition by bringing the power of Transformers to the visual domain. Their ability to extract rich features and capture global context paves the way for even more advanced applications in computer vision.

2.2. BEiT

BEiT models are regular Vision Transformers, but pre-trained in a self-supervised way rather than supervised. Rather than pre-training the model to predict the class of an image (as done in the original ViT paper), BEiT models are pre-trained to predict visual tokens from the codebook of OpenAI's DALL-E model given masked patches.

They outperform both the original model (ViT) as well as Data-efficient Image Transformers (DeiT) when fine-tuned on ImageNet-1K and CIFAR-100.



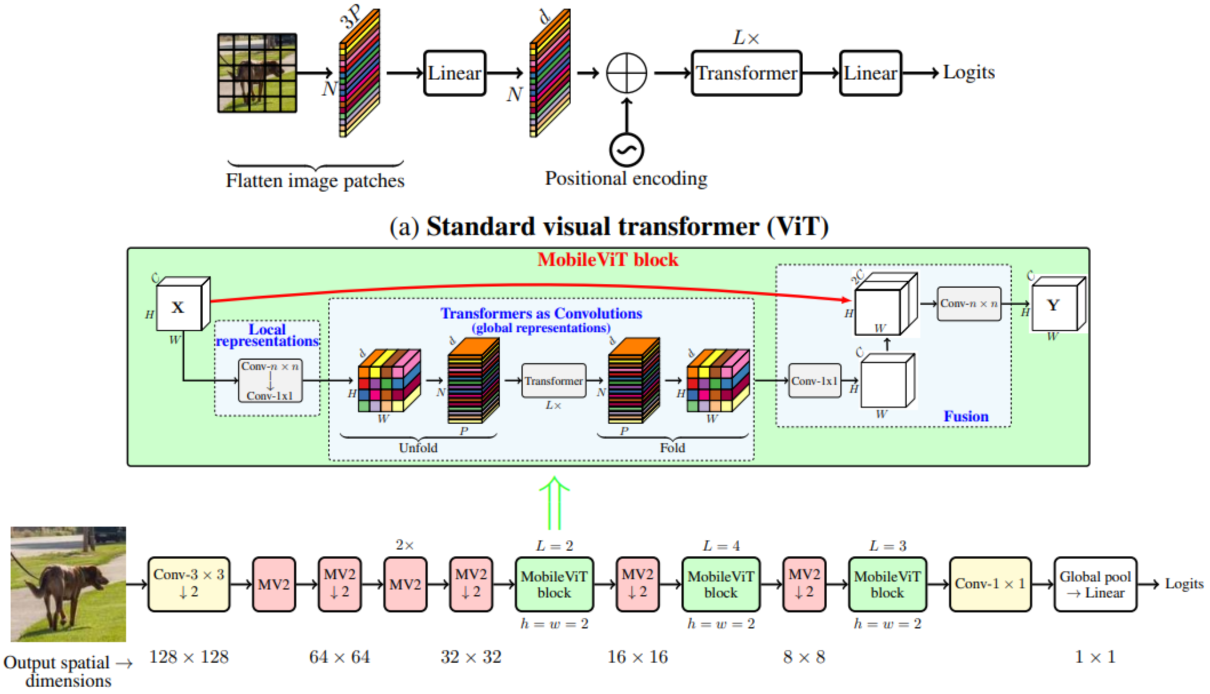
Overview of BEIT pre-training

BEiT uses relative position embeddings, inspired by the T5 model. The pre-training task aims at predicting the visual tokens of the original image based on the encoding vectors of the corrupted image. During pre-training, the authors shared the relative position bias among the several self-attention layers. During fine-tuning, each layer's relative position bias is initialized with the shared relative position bias obtained after pre-training.

BEiT goes beyond simply identifying local features. It extracts a rich tapestry of relationships and context, leading to a deeper understanding of the image and superior performance in various vision tasks.

2.3. MobileViTV2

MobileViT is a light-weight and general-purpose ViT for mobile devices. It presents a different perspective for the global processing of information with Transformers. MobileViT is more like a CNN than a Transformer model. It does not work on sequence data but on batches of images.



(a) **Standard visual transformer (ViT)**

(b) **MobileViT**. Here, Conv-$n \times n$ in the MobileViT block represents a standard $n \times n$ convolution and MV2 refers to MobileNetv2 block. Blocks that perform down-sampling are marked with ↓ 2.

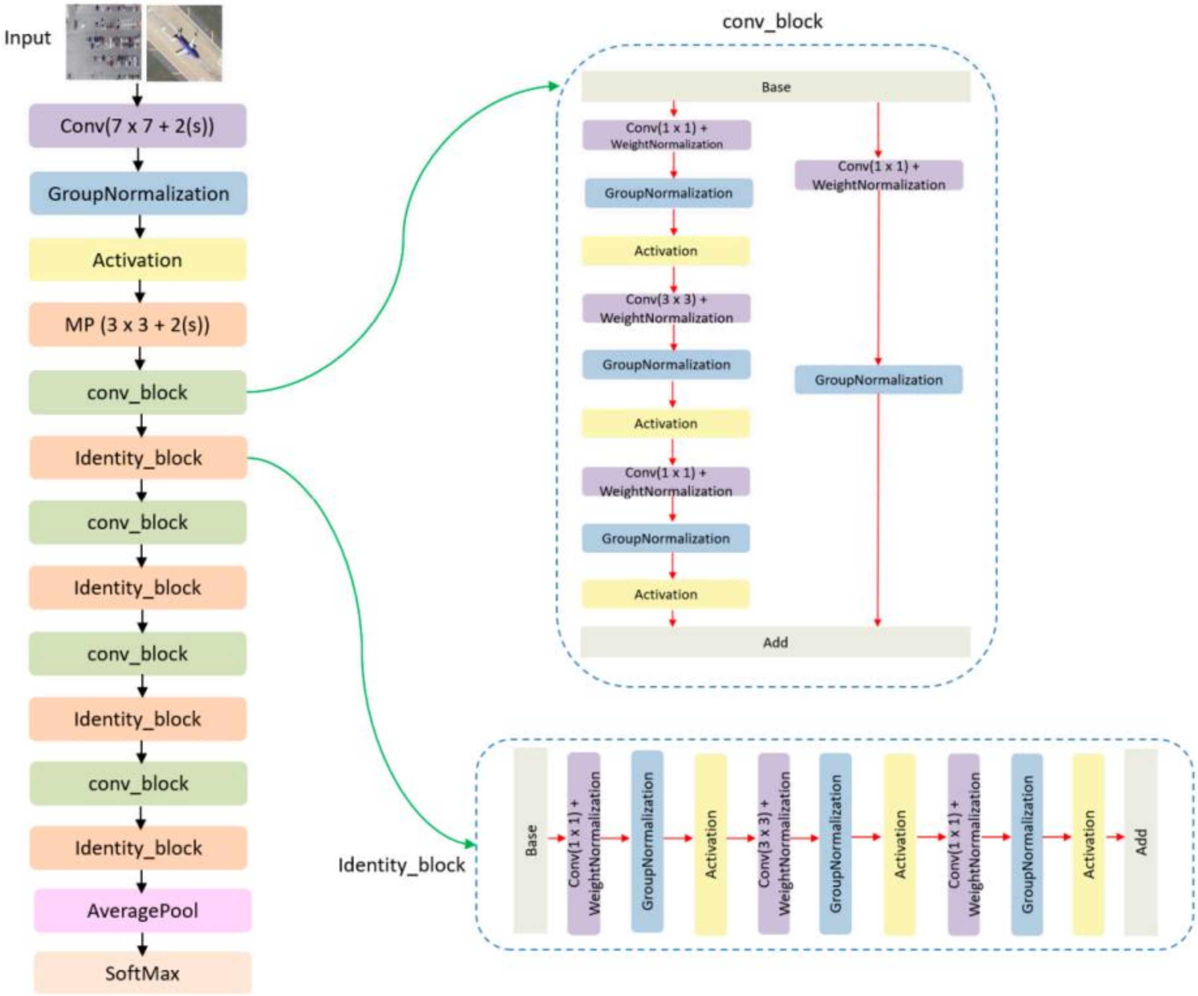
Visual transformers vs. MobileViT

MobileViT block that encodes both local and global information in a tensor effectively. Unlike ViTs, there are no embeddings. Standard convolution involves three operations: unfolding, local processing, and folding. MobileViT block replaces local processing in convolutions with global processing using transformers. This allows MobileViT block to have CNN- and ViT-like properties, which helps it learn better representations with fewer parameters and simple training recipes.

MobileViT represents a significant leap forward in mobile vision by enabling powerful image recognition on devices we carry every day. Its light and efficient architecture opens exciting possibilities for applications like augmented reality, real-time object recognition, and even on-device medical imaging.

2.4. BiT

BiT, standing for Big Transfer, is a pre-training framework for Vision Transformers (ViTs), focusing on scaling up pre-training with a simple and effective recipe. It aims to improve sample efficiency and simplify hyperparameter tuning for training deep neural networks for vision tasks. By pre-training on massive datasets like ImageNet-21k, BiT models learn powerful generic representations that can be fine-tuned for various downstream tasks with impressive results.



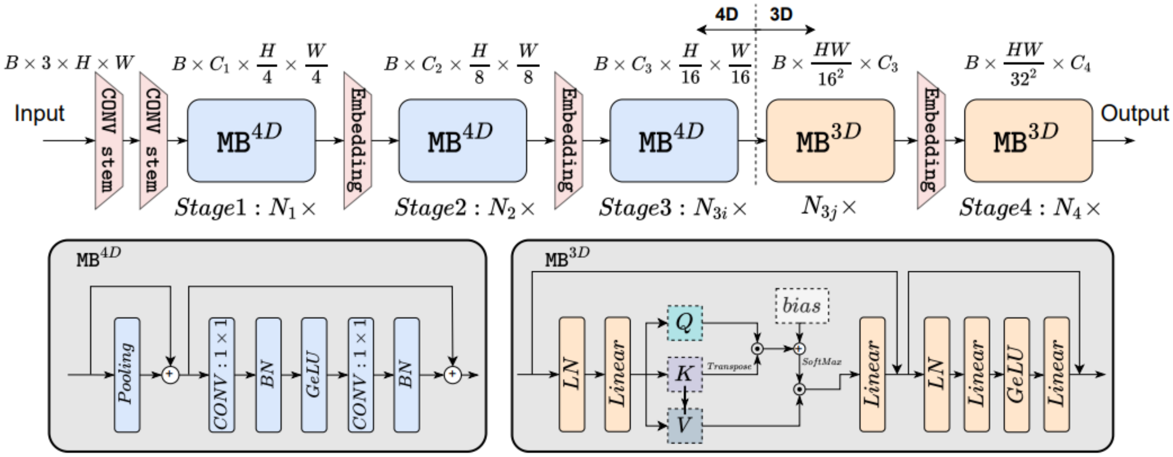
Architecture of Big transfer (BiT) model

BiT has made a significant contribution to the field of image pre-training by demonstrating the value of simplicity and large-scale data in building powerful visual representations. Its impact can be felt across various vision tasks, paving the way for further advancements in computer vision.

2.5. EfficientFormer

Traditionally, mobile vision models relied on lightweight architectures like MobileNets with convolutions for efficiency. EfficientFormer breaks free from this mold by utilizing pure Transformers, a more powerful but resource-intensive architecture. Instead of simply shrinking existing ViT models, EfficientFormer focuses on a "dimension-consistent" design. This means maintaining consistent feature dimensions throughout the model, maximizing information flow and efficiency.

To ensure smooth mobile performance, EfficientFormer employs a clever technique called "latency-driven slimming." It starts with a larger, high-performance model and then systematically prunes parameters and operations based on their impact on inference speed. This ensures optimal accuracy-latency trade-off for each desired performance level.



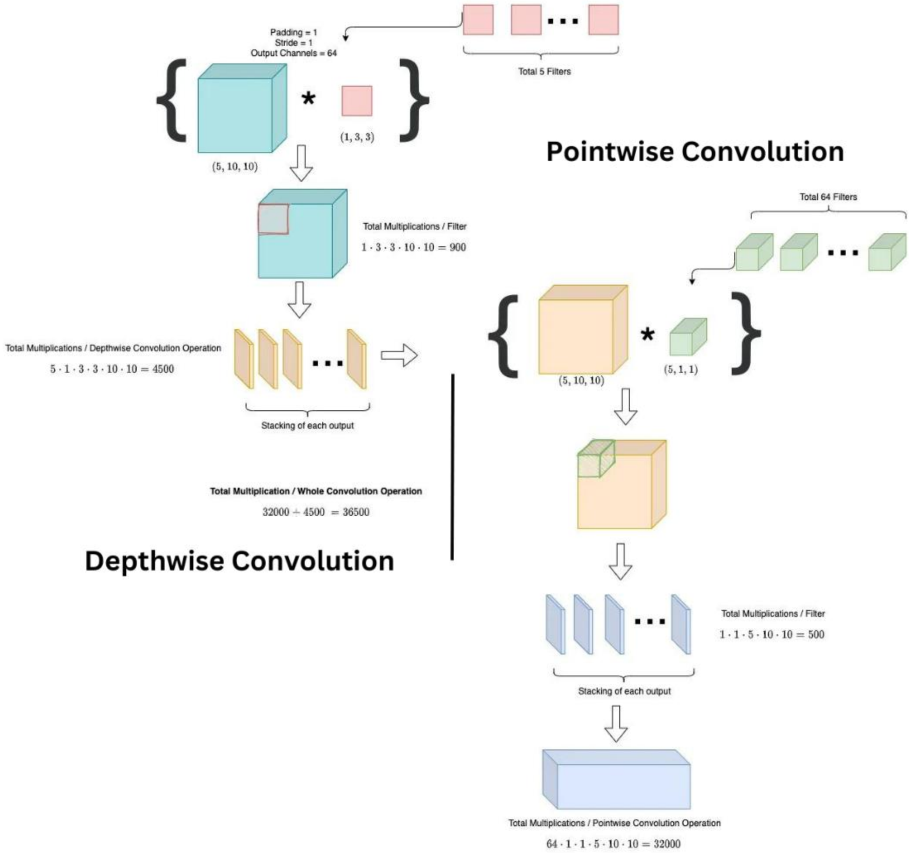
Overview of EfficientFormer

EfficientFormer demonstrates the exciting potential of running powerful Vision Transformers on mobile devices. Its performance and flexibility make it a key player in the future of mobile vision, opening doors to innovative applications and user experiences.

2.6. MobileNetV2

MobileNetV2 was introduced by Google. the paper titled "MobileNetV2: Inverted Residuals and Linear Bottlenecks" was published in 2018. It is an improvement over the original MobileNet, focusing on increased efficiency and better performance.

How MobileNets can reduce several million parameters but still maintain good accuracy is by using a mechanism called Depthwise Separable Convolutional.



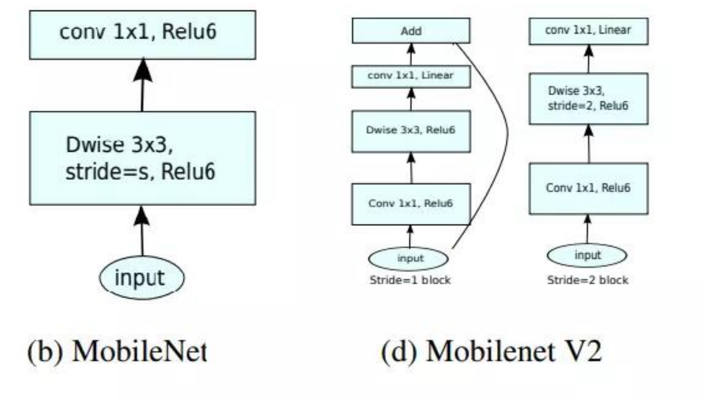
Pointwise Convolution and Depthwise Convolution

Depthwise Separable Convolution is a two-step process that decomposes the standard convolution into depthwise convolution and pointwise convolution.

- Depthwise Convolution: Applies a separate convolutional operation for each input channel (depth-wise operation). It uses a single filter per input channel independently. Produces a set of feature maps, one for each input channel.
- Pointwise Convolution: Applies a 1x1 convolution (cross-channel convolution) to combine the information from depthwise convolution. It uses 1x1 filters and combines the output from depthwise convolution to generate the final output.

Depthwise separable convolution significantly reduces the number of parameters and computations compared to standard convolution. It is computationally more efficient, making it well-suited for mobile and edge devices with limited computational resources.

MobileNetV2 introduces inverted residuals, linear bottlenecks, shortcut connections, and global average pooling to achieve a good balance between accuracy and computational efficiency.



(b) MobileNet (d) Mobilenet V2

The differences between v1 and v2

Analysis of the differences in the provided image: MobileNetV2 uses two types of blocks, including a residual block with a stride of 1 and a block with a stride of 2 for downsizing. There are three parts for each block:

- The first layer is a 1×1 convolution with ReLU6.
- The second layer, as before, is depthwise convolution.
- The third layer continues with 1×1 convolution but without an activation function. Linear is used instead of ReLU as usual.

Connection shortcuts in MobileNetV2 are adjusted so that the number of input and output channels for each residual block is narrowed down. Hence, they are called bottleneck layers.

The residual block of MobileNetV2 is contrary to traditional residual architectures, as the traditional residual architecture has a larger number of channels in the input and output of a block compared to intermediate layers. Therefore, it is also called an Inverted Residual Block.

The intermediate layers in a block will perform non-linear transformations, so they need to be thicker to generate more transformations. Shortcut connections between blocks are implemented on bottleneck inputs and outputs rather than on intermediate layers. Therefore, bottleneck input and output layers only need to record the result and do not need to perform non-linear transformations (linear f).

Between layers in an inverted residual block, we also use separate Depthwise Separable Convolutions to minimize the number of model parameters. This is the secret that helps this model have a reduced size.

2.7. ResNet

ResNet, short for Residual Networks, was proposed by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. The paper titled "Deep Residual Learning for Image Recognition" was presented at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in 2016.

ResNet addresses the challenge of training very deep neural networks, Vanishing Gradient. In reality, Gradients will often have gradually smaller values as they go down to lower layers. As a result, the updates performed by Gradients Descent do not change the weights of those layers much and they cannot converge and the network will not get good results. Such a phenomenon is called Vanishing Gradients.

In order to solve the problem of the vanishing/exploding gradient, this architecture introduced the concept called Residual Blocks. In this network, we use a technique called skip connections. The skip connection connects activations of a layer to further layers by skipping some layers in between. This forms a residual block. Resnets are made by stacking these residual blocks together.

The approach behind this network is instead of layers learning the underlying mapping, we allow the network to fit the residual mapping. So, instead of say H(x), initial mapping, let the network fit.

$$F(x) = H(x) - x \text{ which gives } H(x) = F(x) + x$$

The advantage of adding this type of skip connection is that if any layer hurt the performance of architecture, then it will be skipped by regularization. So, this results in training a very deep neural network without the problems caused by vanishing/exploding gradient. The authors of the paper experimented on 100-1000 layers of the CIFAR-10 dataset.

There is a similar approach called “highway networks”, these networks also use skip connection. Like LSTM these skip connections also use parametric gates. These gates determine how much information passes through the skip connection. This architecture however has not provided accuracy better than ResNet architecture.

2.8. EfficientNet

EfficientNet is a revolutionary deep neural network architecture designed to address optimization challenges in the realm of model scaling. The core idea behind EfficientNet's Model Scaling is to strike a balance between three key factors: Depth Scaling, Width Scaling, and Resolution Scaling:

- Depth Scaling: EfficientNet increases the depth of the network, allowing it to capture more complex features. However, excessively deep networks can suffer from vanishing gradients and increased computational cost.
- Width Scaling: EfficientNet adjusts the width of the network, controlling the number of channels in each layer. While wider networks can capture more information, they may also require more resources.
- Resolution Scaling: EfficientNet varies the resolution of the input images, enabling the model to adapt to different levels of detail. However, higher resolutions demand increased computational power.

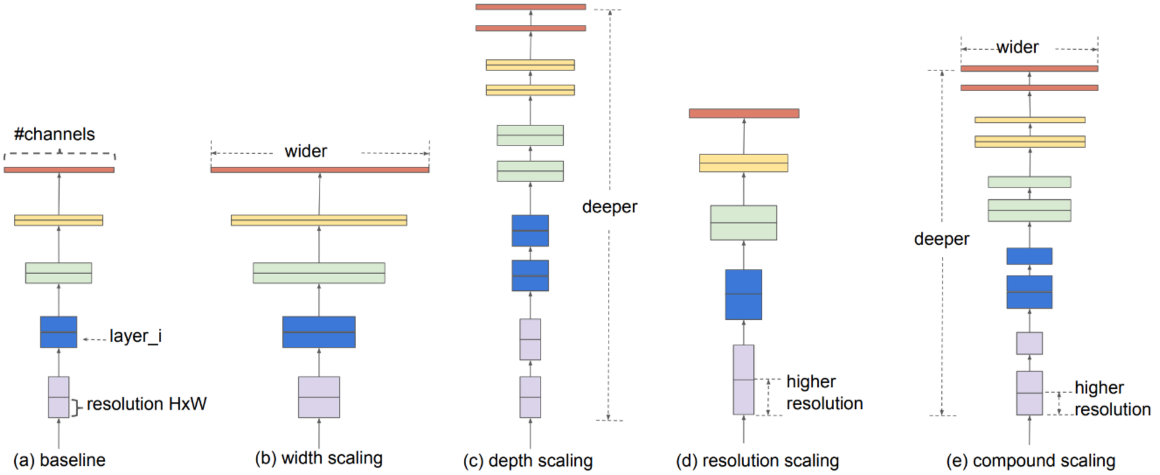


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

EfficientNet introduces the concept of Compound Scaling, a sophisticated method that uniformly scales the depth, width, and resolution across all layers of the network. This approach optimizes the model's efficiency and performance while addressing the shortcomings of individual scaling methods.

CNN networks can be integrated as overlapping convolutional layers. Furthermore, these layers can be divided into different stages (layers), so it be represented mathematically as follows:

$$N = \bigodot_{i=1...s} F_i^{L_i}(X_{<H_i,W_i,C_i>})$$

In there N describe the neural network, i represents the number of stages, F_i demonstrate convolution operation for i-th stage and L_i perform number of times F_i is repeated in the i-th stage. H_i, W_i, C_i describe the input tensor shape for the i-th stage in turn.

Along with some of the mentioned principles, the above equation is parameterized to solve the optimization problem as follows:

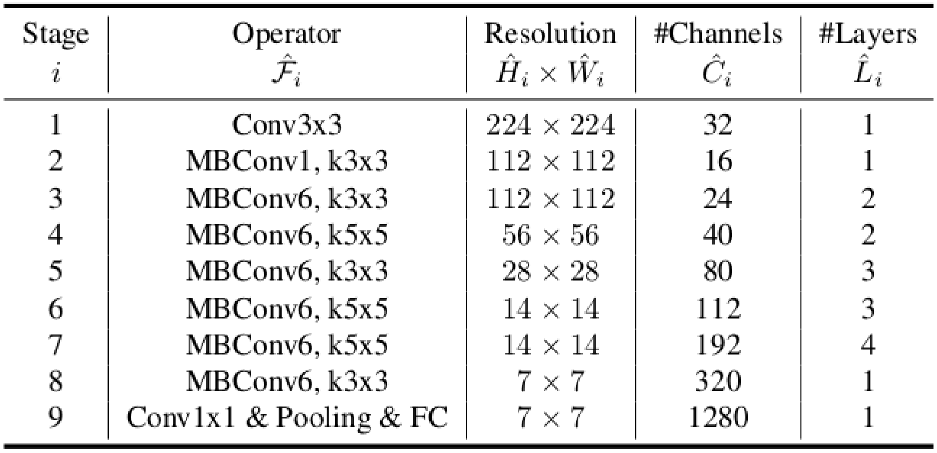
$$N(d,w,r) = \bigodot_{i=1\ldots s} \hat{F}_i^{d\cdot\hat{L}_i}\left(X_{<r\cdot\hat{H}_i, r\cdot\hat{W}_i, r\cdot\hat{C}_i>}\right)$$

In there w, d, r are the network expansion coefficients corresponding to width, depth and resolution. $\widehat{F_\iota}$, $\widehat{L_\iota}$, $\widehat{H_\iota}$, $\widehat{W_\iota}$, $\widehat{C_\iota}$ are predefined parameters in the base network.

Compound coefficient to evenly expand the NN network in width, depth and resolution in a disciplined way:

$$\text{depth}: d = \alpha^{\phi} \ \text{width}: w = \beta^{\phi} \ \text{resolution}: r = \gamma^{\phi} \ s.t \quad \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2, \alpha \geq 1, \beta \geq 1, \gamma \geq 1$$

To better demonstrate the effectiveness of their scaling approach, the authors also developed a cellular base-scale network, called EfficientNet-B0.

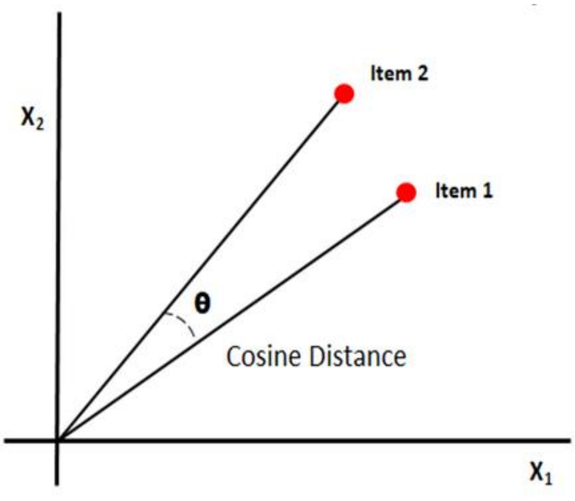


Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

3. Similarity Calculation

3.1. Cosine Distance

Cosine similarity quantifies the similarity between two vectors by measuring the cosine of the angle formed between them. The more similar the vectors, the closer the similarity value approaches 1 (indicating a smaller angle). Hence, the Cosine distance is defined as 1 minus the cosine of the angle between the two vectors.



With two feature vectors u and v, the cosine distance between them is calculated by the formula:

$$\boldsymbol{Similarity = 1 - cos(\theta) = \frac{u.v}{\|u\|\|v\|}}$$

Yet, Cosine similarity solely computes the angle between two vectors, focusing on direction rather than magnitude. This approach results in information loss and consequently diminishes overall performance.

3.2. Euclide Distance

Euclidean Distance stands out as the most frequently employed and straightforward technique in practical applications. It is utilized for determining the length distance between two points connected by a line segment, commonly denoted as the L-2 distance.

The formula calculating Euclidean Distance:

$$\boldsymbol{D(u,v) = \sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}}$$

Euclidean Distance proves highly efficient for data with lower dimensions due to its simplicity, ease of comprehension, and ability to yield satisfactory results. Nonetheless, its effectiveness is influenced by both the feature unit and the vector's dimensionality. Therefore, in practical applications dealing with extensive and multidimensional datasets, the efficiency of Euclidean Distance diminishes.

3.3. TS-SS Similarity

Problems with Euclidean Distance: If two data vectors have no component values in common, they may have a smaller distance than the other pair of data vectors containing the same component values.

Problems with Cosine Similarity: The major disadvantage with cosine similarity is that it does not take magnitude into account. We have seen that cosine similarity is directly proportional to the inner product of the two vectors but do not consider the difference in their lengths it’s just only focused on orientation.

TS-SS computes the similarity between two vectors from diverse perspective and generates the similarity value from two vectors not only from the angle and Euclidean distance between them, but also the difference between their magnitudes.

TS: Triangle Area Similarity includes three major characteristics for computing similarity:

- Angle
- Euclidean Distance between vectors
- Magnitude of vectors

The intuition behind the usage of TS is when we draw the Euclidean distance line between two vectors as we have in the figure above we can clearly see that if vectors are closer ED is small, the angle between vectors also changes and even the are of triangle decreases. So all these factors got affected if two vectors come close to each other. It’s wise to incorporate them together to create a metric component.

$$\boldsymbol{TS(A,B)} = \frac{\boldsymbol{|A| \cdot |B| \cdot \sin(\theta')}}{\boldsymbol{2}}$$

SS: Sector Area Similarity, TS will not be able to completely solve our problem because like cosine similarity it is missing an important component which is the difference in magnitude of vectors. TS can be understood as an enhancement to cosine similarity, but we also must combine Euclidean distance, which will be covered in this part.

$$\boldsymbol{MD(A,B)} = \left| \sqrt{\sum_{n=1}^{k} A_n^2} - \sqrt{\sum_{n=1}^{k} B_n^2} \right|$$

$$\boldsymbol{SS(A,B)} = \boldsymbol{\pi} \cdot \left(\boldsymbol{ED(A,B)} + \boldsymbol{MD(A,B)}\right)^2 \cdot \left(\frac{\boldsymbol{\theta'}}{\boldsymbol{360}}\right)$$

The range of TS_SS measure from 0 to ∞.

The reason for choosing multiplication but not summation TS and SS is that in some cases the value of TS and SS are disproportionate where one is extremely larger than the other one.

4. System Evaluation

In the context of evaluating a model in information retrieval, various metrics, namely *Average Precision (AP), Mean Reciprocal Rank (MRR), Non-interpolated Mean Average Precision (MAP), and Interpolated MAP*, are employed. The dataset comprises 110 pre-assigned ground truth query images categorized as Good, OK, Junk, and Bad based on the clarity of object views. For evaluation, the "good" and "ok" files are combined, considering each image returned by the system in both files as "relevant," and none as "irrelevant".

Mean Reciprocal Rank (MRR) is a statistic measure for evaluating any process that produces a list of possible responses to a sample of queries, ordered by probability of correctness. The reciprocal rank of a query response is the multiplicative inverse of the rank of the first correct answer: 1 for first place, 1⁄2 for second place, 1⁄3 for third place and so on. The mean reciprocal rank is the average of the reciprocal ranks of results for a sample of queries Q:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}$$

Where $\boldsymbol{rank_i}$ refers to the rank position of the *first* relevant document for the *i*-th query.

Mean Average Precision (mAP) is utilized to assess the system's performance and compare different methods. It is a measure employed in information retrieval systems, evaluating performance in querying information by sorting related documents based on relevance.

The calculation of mAP involves precision and recall values. Precision is the ratio of related documents returned to the total documents returned, while recall is the ratio of related documents returned to the total number of related documents. Average precision is calculated as the average of precision at positions where the return is deemed relevant.

$$AP = \frac{\sum_{k=1}^{n} P(k) * rel(k)}{number\ of\ relevant\ documents}$$

$$mAP = \frac{\sum_{q=1}^{Q} AP(q)}{Q}$$

Non-interpolated mAP (MAP) is computed as the average of average precision across all queries in the dataset. Additionally, an interpolated mAP is carried out using 11-point interpolated average precision. This involves measuring interpolated precision at 11 recall levels (0.0, 0.1, 0.2, ..., 1.0) for each information need, and the arithmetic mean of interpolated precision at each level is calculated. The interpolated precision at a given recall level r is defined as the highest precision found for any recall level $r' \geq r$:

$$p_{inter}(r) = \max_{r' \geq r} p(r')$$

5. Result

After conducting 110 queries, we analyzed the performance of an object retrieval system with the following results:

Regarding to The Oxford Building Dataset:

Method	Cosine Evaluation				Euclidean Evaluation				TS_SS Evaluation			
	Non-interpolated MAP	Interpolated MAP	Mean Reciprocal Rank	Time (s)	Non-interpolated MAP	Interpolated MAP	Mean Reciprocal Rank	Time (s)	Non-interpolated MAP	Interpolated MAP	Mean Reciprocal Rank	Time (s)
BEiT	0.83	0.55	0.95	13.43	0.83	0.53	0.93	10.77	0.79	0.51	0.89	29.59
BiT	0.67	0.38	0.81	11.5	0.68	0.38	0.83	9.21	0.67	0.38	0.83	27.78
EfficentFormer	0.59	0.29	0.75	7.35	0.56	0.27	0.73	8.42	0.55	0.27	0.71	22.92
EfficentNet	0.75	0.39	0.92	15.32	0.74	0.38	0.91	14.18	0.75	0.39	0.91	32.11
MobileNetV2	0.59	0.31	0.77	10.44	0.57	0.29	0.75	7.04	0.60	0.28	0.76	26.25
MobileViTV2	0.65	0.31	0.82	9.83	0.63	0.27	0.76	7.78	0.62	0.28	0.78	24.00
ResNet	0.65	0.3	0.83	9.04	0.64	0.28	0.84	9.11	0.65	0.3	0.82	26.11
VisionTranformer	0.70	0.43	0.84	9.94	0.70	0.43	0.83	9.28	0.69	0.42	0.82	26.33

Regarding to Paris Building Dataset:

Method	Cosine Evaluation				Euclidean Evaluation				TS_SS Evaluation			
	Non-interpolated MAP	Interpolated MAP	Mean Reciprocal Rank	Time (s)	Non-interpolated MAP	Interpolated MAP	Mean Reciprocal Rank	Time(s)	Non-interpolated MAP	Interpolated MAP	Mean Reciprocal Rank	Time (s)
BEiT	0.91	0.61	1.00	15.31	0.91	0.66	1.00	13.66	0.90	0.63	1.00	35.81
BiT	0.87	0.61	0.98	14.70	0.87	0.97	0.97	12.34	0.87	0.60	0.98	34.73
EfficentFormer	0.84	0.98	0.98	10.53	0.84	0.99	0.99	8.00	0.83	0.97	0.97	27.15
EfficentNet	0.85	0.99	0.99	16.94	0.85	0.99	0.99	15.36	0.84	0.99	0.99	36.85
MobileNetV2	0.84	0.98	0.98	10.27	0.84	0.51	0.95	8.70	0.84	0.50	0.99	30.14
MobileViTV2	0.85	0.53	1.00	11.95	0.84	0.49	1.00	9.03	0.83	0.48	1.00	29.55
ResNet	0.84	0.53	0.98	10.87	0.84	0.51	0.98	8.92	0.83	0.50	0.98	31.15
VisionTranformer	0.88	0.62	1.00	12.35	0.88	0.61	1.00	10.84	0.87	0.61	1.00	31.94

Regarding to all dataset (combined dataset):

Method	Cosine Evaluation				Euclidean Evaluation				TS_SS Evaluation			
	Non-interpolated MAP	Interpolated MAP	Mean Reciprocal Rank	Time (s)	Non-interpolated MAP	Interpolated MAP	Mean Reciprocal Rank	Time(s)	Non-interpolated MAP	Interpolated MAP	Mean Reciprocal Rank	Time (s)
BEiT	0.86	0.56	0.97	50.23	0.86	0.59	0.96	43.48	0.84	0.56	0.95	128.84
BiT	0.76	0.48	0.89	40.09	0.76	0.47	0.89	33.68	0.76	0.47	0.90	101.04
EfficentFormer	0.70	0.39	0.86	30.75	0.69	0.36	0.85	25.93	0.68	0.36	0.83	87.80
EfficentNet	0.78	0.44	0.95	49.33	0.77	0.42	0.93	42.91	0.78	0.42	0.94	114.66
MobileNetV2	0.70	0.40	0.86	34.89	0.71	0.37	0.85	27.82	0.71	0.37	0.86	95.92
MobileViTV2	0.74	0.40	0.90	32.71	0.73	0.36	0.87	27.40	0.72	0.36	0.87	93.80
ResNet	0.73	0.39	0.89	38.27	0.73	0.37	0.91	32.73	0.73	0.36	0.88	103.81
VisionTranformer	0.78	0.51	0.91	37.96	0.77	0.50	0.91	32.47	0.77	0.50	0.90	104.03

According to the results, BeiT has archived the best performance of all methods in term of accuracy and return relevant images effectively.

ViT-based models like ViT, BeiT, BiT, and MobileViTV2 excel at capturing complex relationships and are good choices for diverse and detail-rich datasets, but computationally expensive.

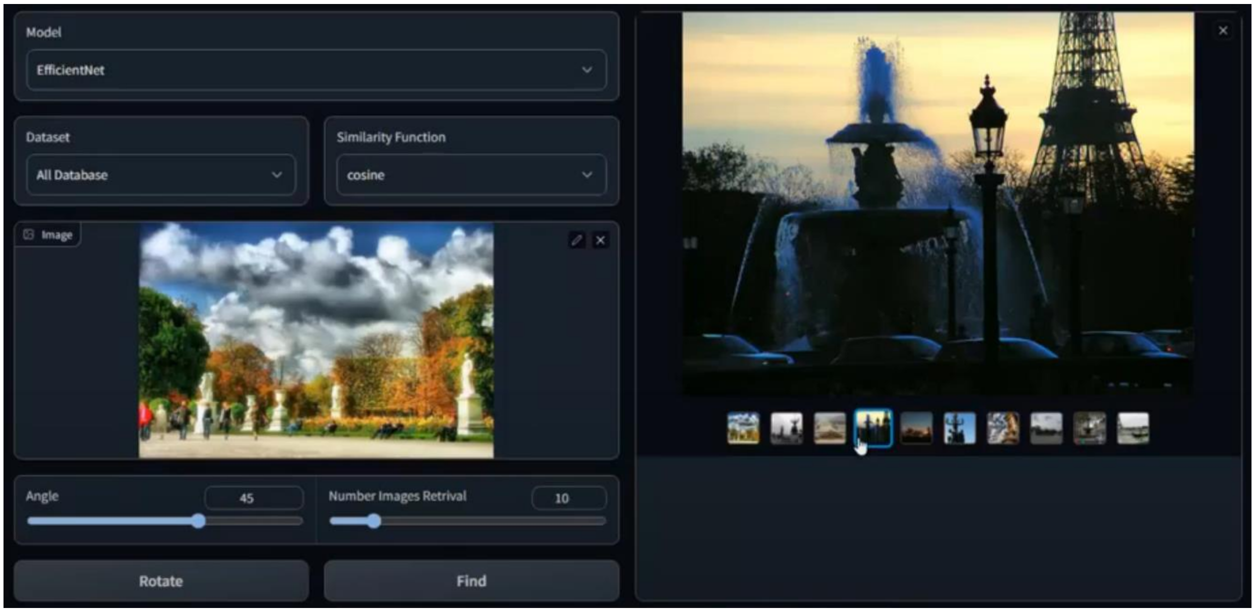
EfficientFormer and EfficientNet offer a good balance between accuracy and efficiency.

MobileNetV2, a lightweight model is ideal for real-time applications with large image collections, but MobileViTV2 is an alternative with better peformance.

ResNet can be a viable option for initial explorations or resource-constrained scenarios. However, if high accuracy and understanding of complex relationships are paramount, newer models is more suitable.

III. DEMO

Our system is built using Gradio framework. Unlike traditional image search methods relying on keywords, our system empowers users to search for visually similar images through a user-friendly interface.



Model: EfficientNet

Dataset: All Database

Similarity Function: cosine

Angle: 45

Number Images Retrival: 10

Screen shot from video

We have recorded our demonstration as video: Drive

Source and app is compressed and attached with this report. Backup: Drive

IV. CONCLUSION

In summary, image retrieval is like a smart search engine for pictures. Our project aimed to make finding and organizing images faster and easier. We used advanced computer techniques to create a system that can recognize and retrieve images effectively.

We tested our system using specific datasets, like a collection of Oxford and Paris building images, and it worked well. But we didn't stop there; we also made our system accessible through a website, allowing users to easily search for images.

Looking ahead, we plan to improve our system even more. We'll explore the latest computer technologies to make our system better at recognizing and finding images quickly. This includes using advanced methods to compare images and smart techniques to handle large image databases. Our goal is to make image searching not only efficient but also user-friendly for everyone.

REFERENCES

[1] Vision Transformers (ViT) in Image Recognition – 2024 Guide

[2] Vision Transformer and MLP-Mixer Architectures

[3] When Vision Transformers Outperform ResNets without Pre-training or Strong Data Augmentations

[4] An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

[5] BEiT: BERT Pre-Training of Image Transformers

[6] MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer

[7] Big Transfer (BiT): General Visual Representation Learning

[8] EfficientFormer: Vision Transformers at MobileNet Speed

[9] MobileNetV2: Inverted Residuals and Linear Bottlenecks

[10] Deep Residual Learning for Image Recognition

[11] EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks