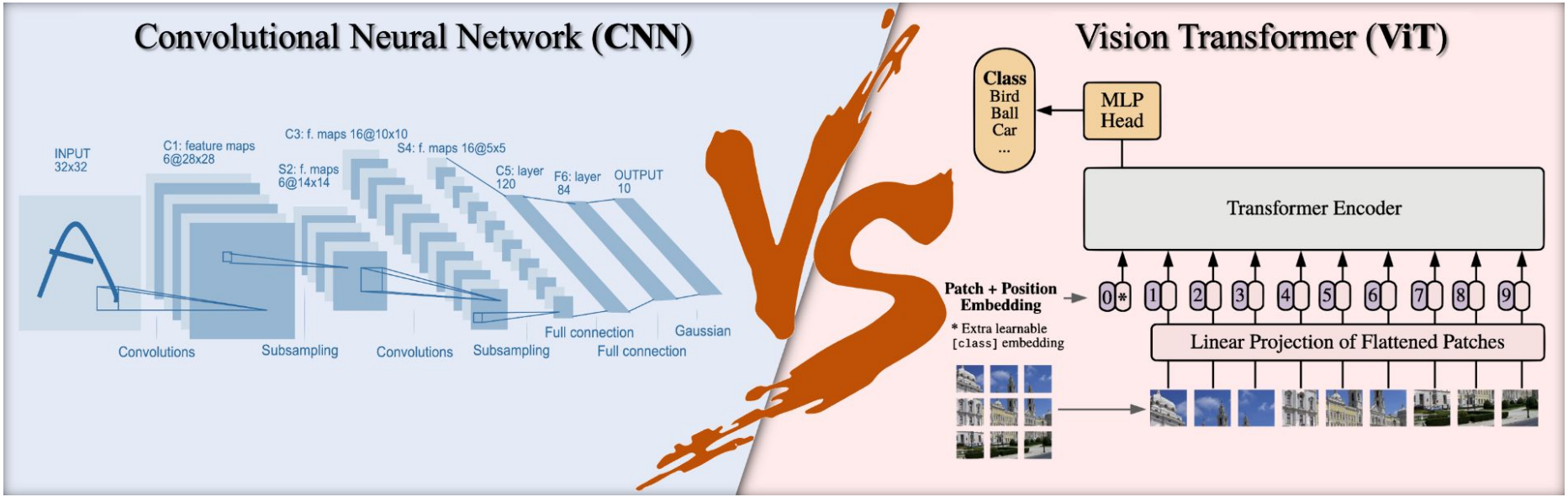
Vision Transformer

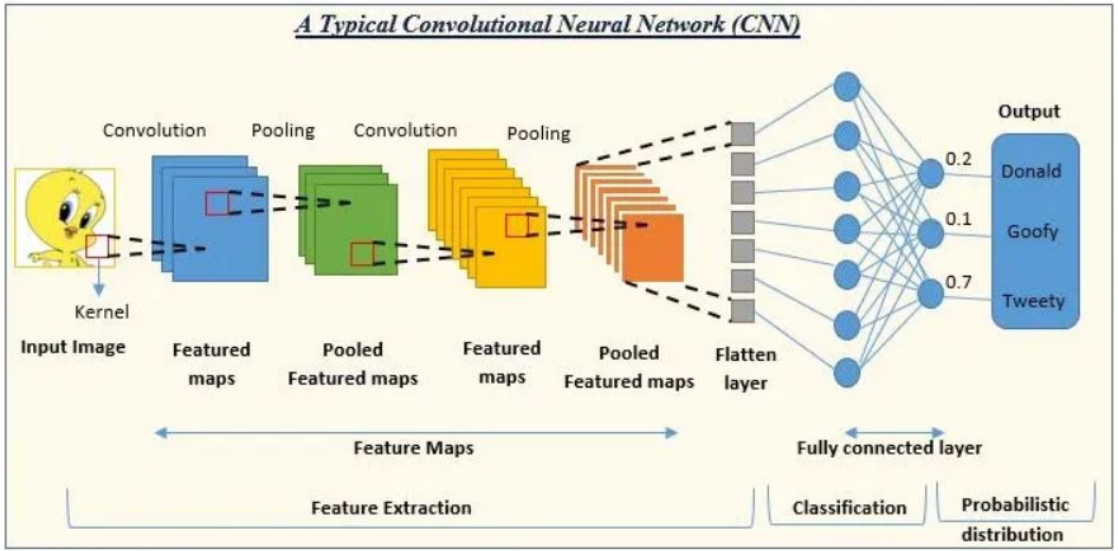


TA Hùng An

Nội dung

1. Introduction
2. Transformer vs CNN
3. Mean Teacher

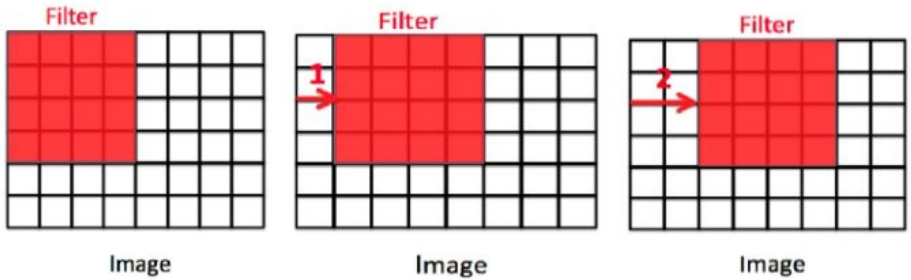
1 - Introduction



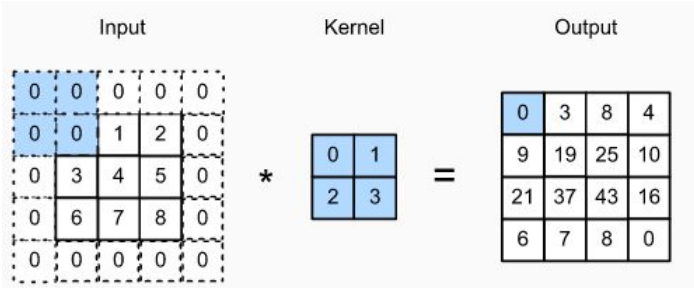
CNN là kiến trúc mạng được sử dụng rộng rãi cho một số tác vụ liên quan đến visual recognition bằng sự thành công của phép tích chập (convolutional operation)

1 - Introduction

- Stride và Padding cũng là những nhân tố tác động đến tính hiệu quả của CNN
- Stride - số bước mà kernel di chuyển
- Padding - thêm vùng biên để tăng kích thước ma trận



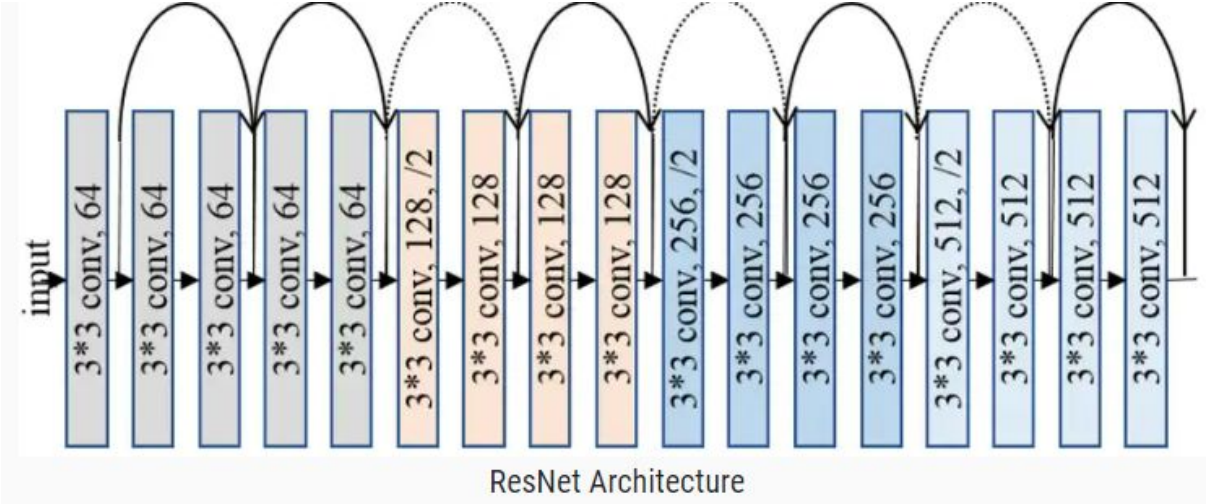
Strides



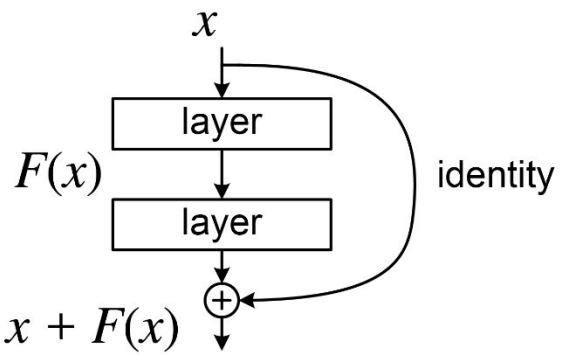
Padding

1 - Introduction

- ResNet là một kiến trúc mạng tiêu biểu và thịnh hành trong nhóm CNN.
- Ý tưởng Skip connection của ResNet được ứng dụng rất nhiều cho một số model Architecture sau này.

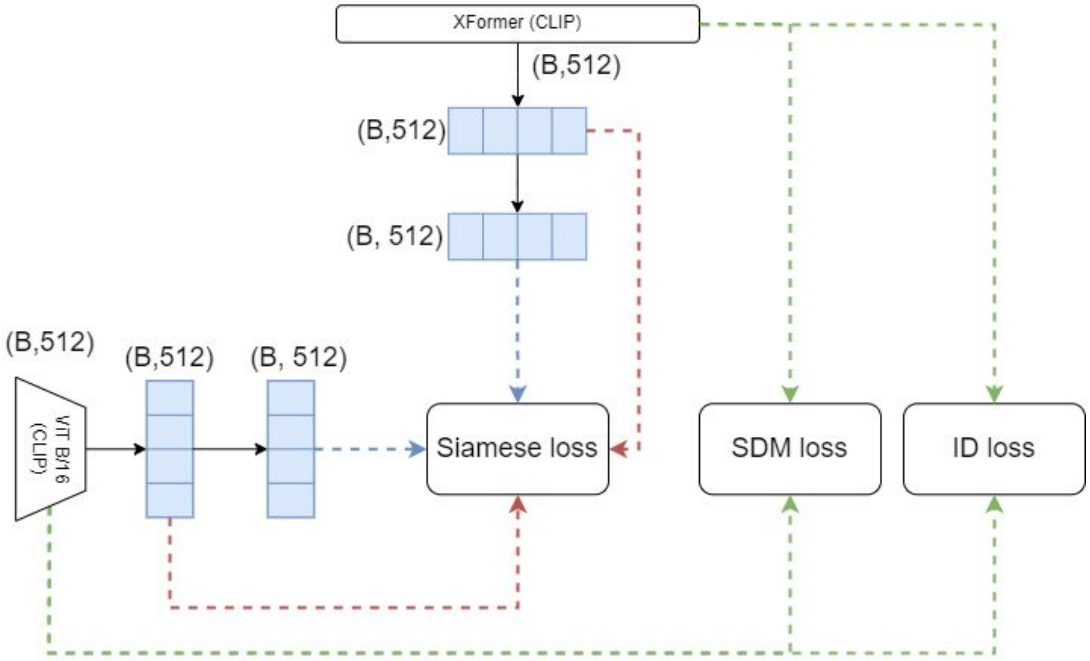


ResNet Architecture



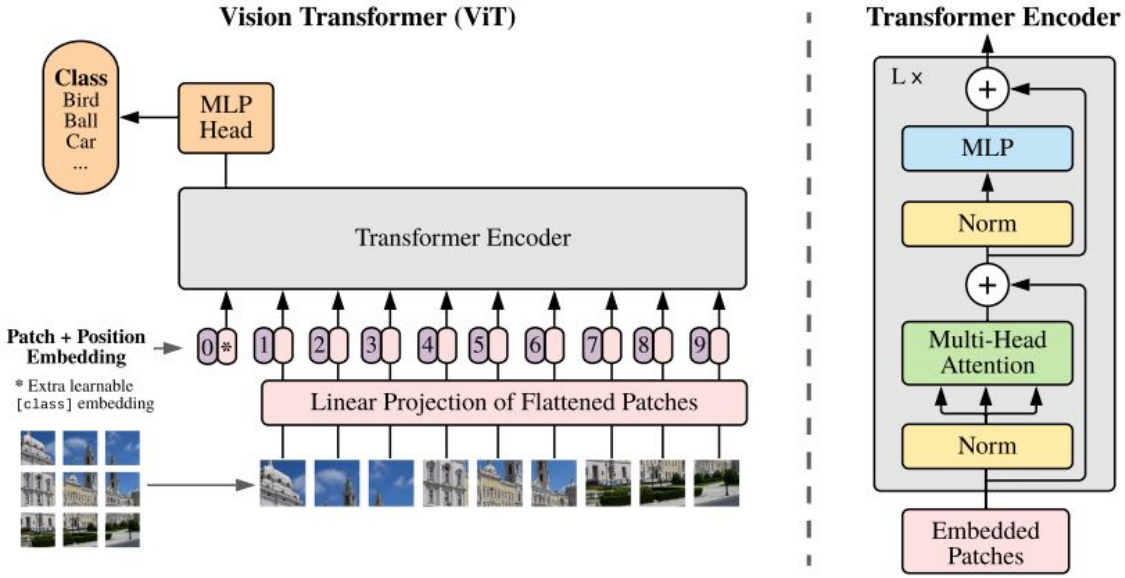
1 - Introduction

Ứng dụng Skip Connection cho một model Architecture phục vụ cho bài toán Multi Task Learning



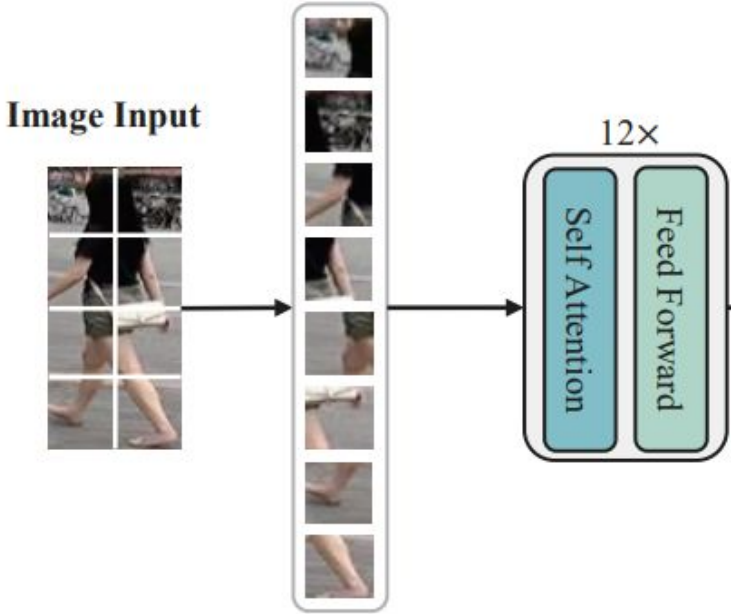
1 - Introduction

Sự thống trị của CNN càng bị thách thức bởi Transformer khi các nhà nghiên cứu đã áp dụng thành công một architecture chuyên xử lý cho bài toán NLP vào bài toán CV



1 - Introduction

Chúng ta có thể thay đổi cơ chế chia patch của ViT tùy theo mục đích sử dụng hay bối cảnh bài toán.

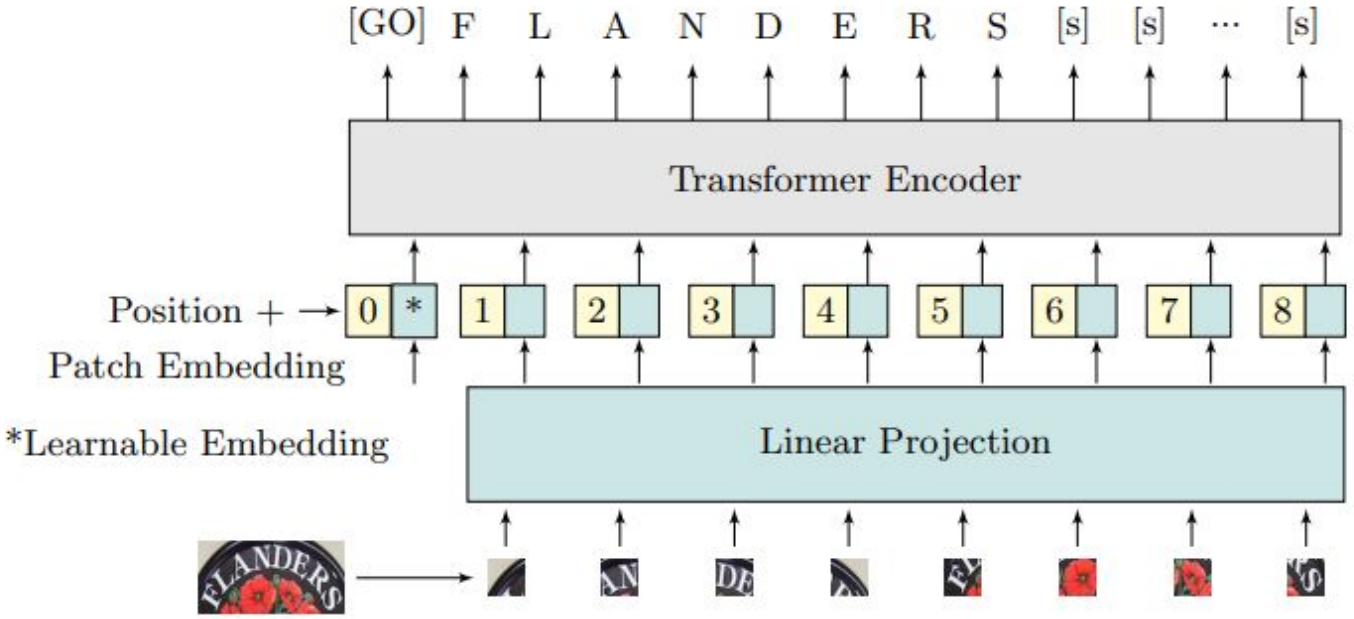



```
# custom patch
'''
self.patch_size = (image_size//4, image_size//2)
self.n_patches = (image_size // self.patch_size[0])*((image_size // self.patch_size[1]))
'''

# convolutional layer that does both the splitting into patches and their embedding
self.cv2D_layer = nn.Conv2d(in_channels, embed_dims, kernel_size=patch_size, stride=patch_size)
```

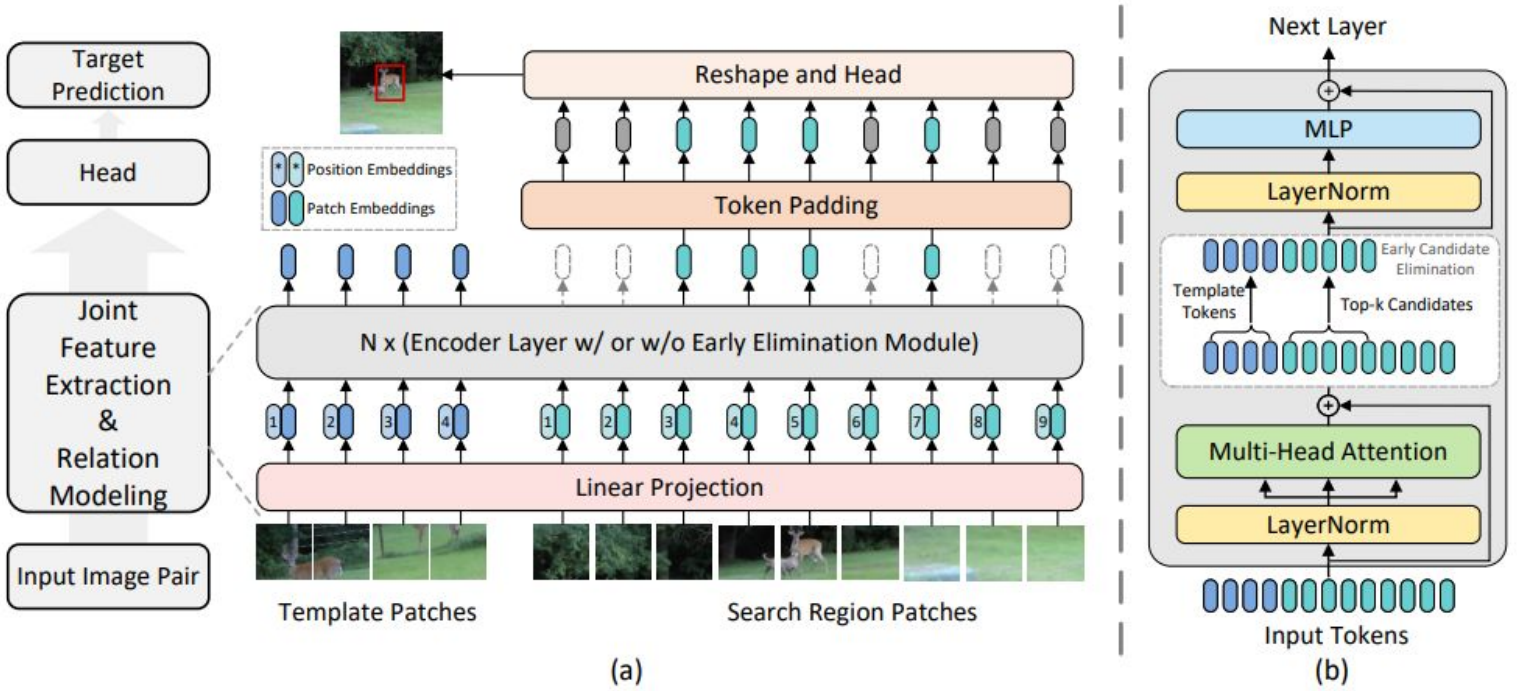
1 - Introduction

Ứng dụng Transformer cho bài toán Text Recognition



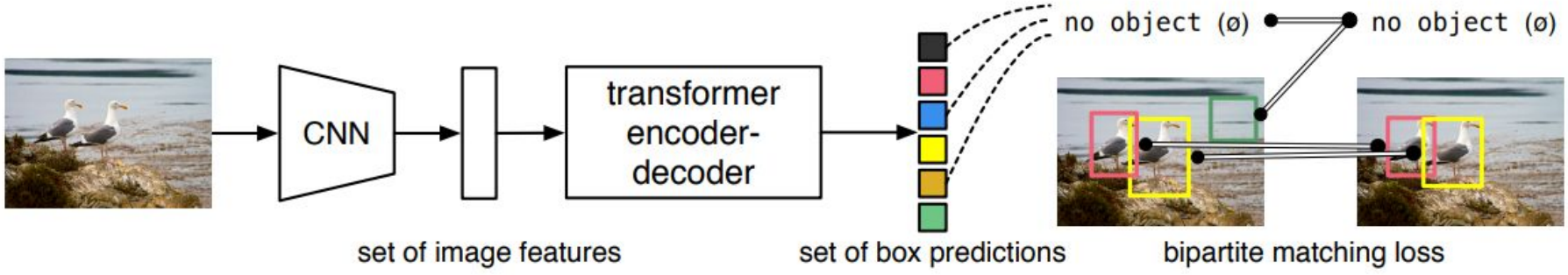
1 - Introduction

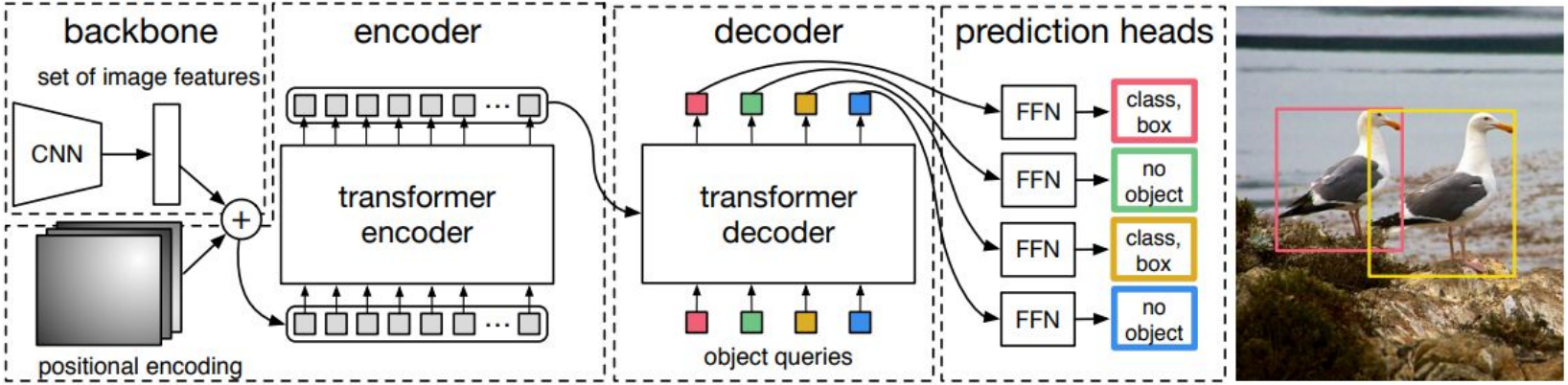
Ứng dụng Transformer cho bài toán Visual Object Tracking



1 - Introduction

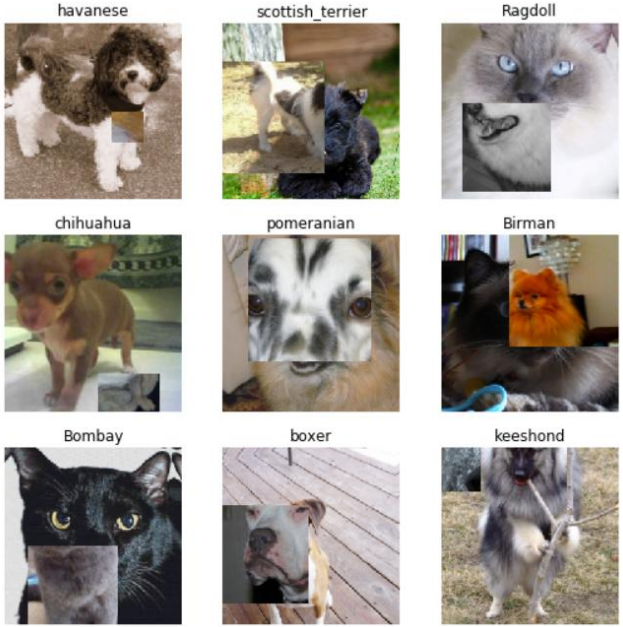
Ứng dụng Transformer cho bài toán Object Detection





2 - Transformer vs CNN

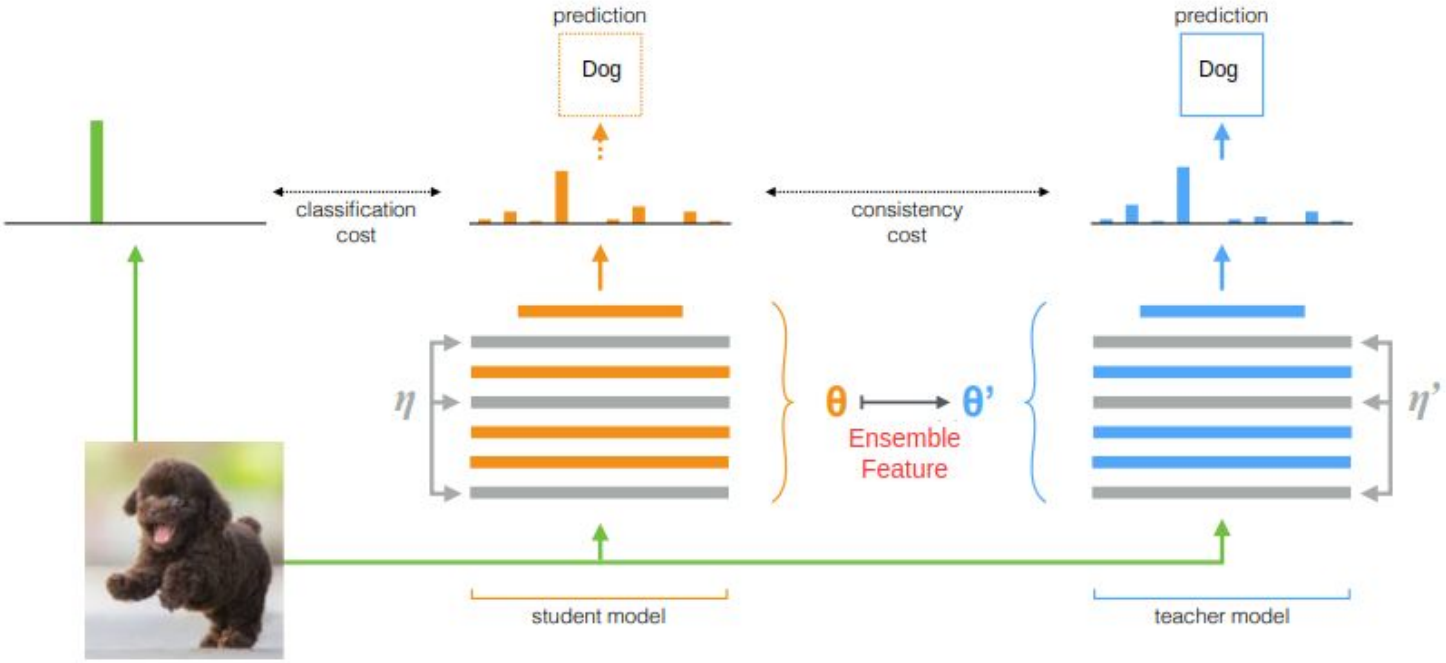
Áp dụng Strong Augmentation



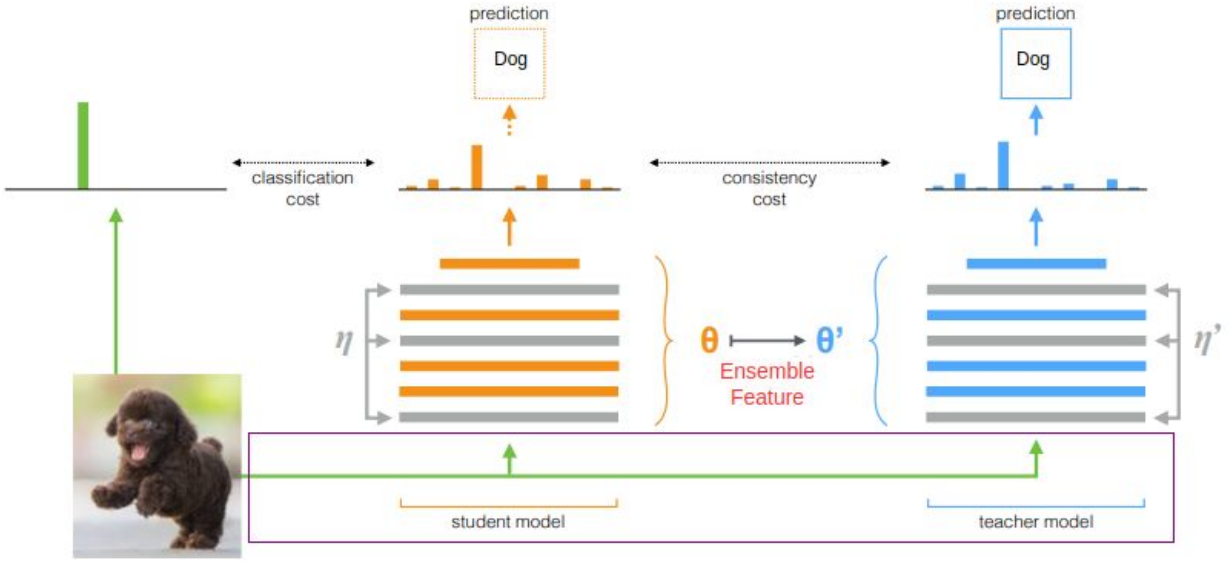
Cutmix

Mixup

3 - Mean Teacher

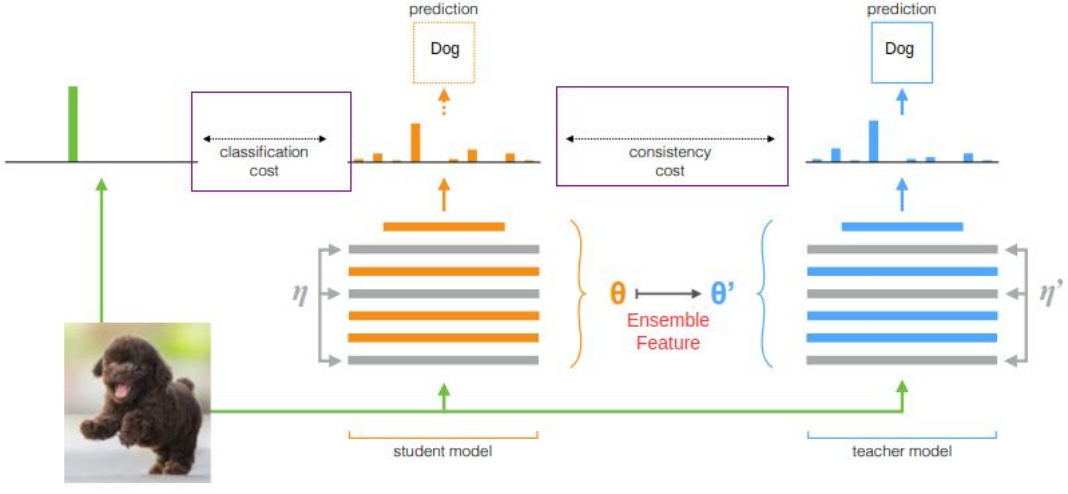


3 - Mean Teacher




```
# forward pass
student_pred = student(images)
teacher_pred= teacher(images)
```

3 - Mean Teacher

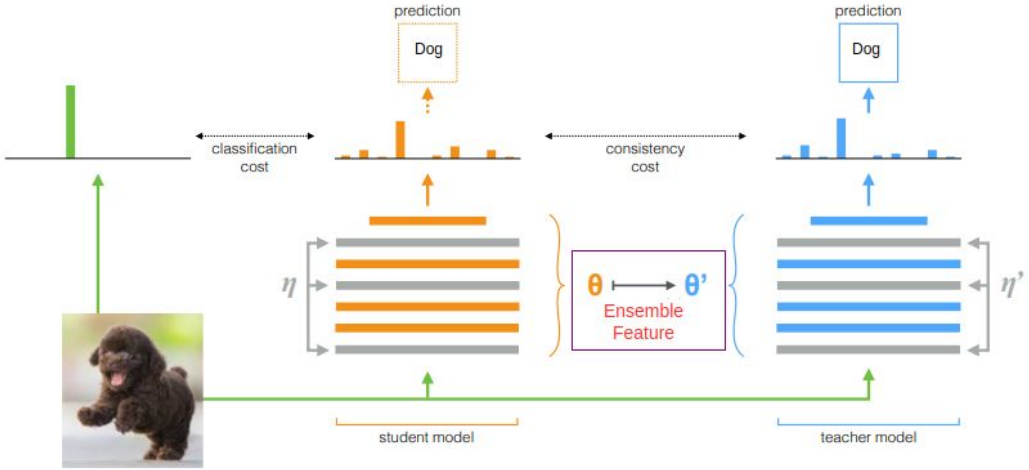



```
student_class, student_consistency = student_pred, student_pred

student_class_loss = class_criterion(student_class, labels) # CrossEntropy
consistency_loss = consistency_criterion(student_consistency, teacher_pred) # MSE

loss = student_class_loss + consistency_loss
```

3 - Mean Teacher



$$\theta'_t = \alpha\theta'_{t-1} + (1-\alpha)\theta_t$$

```
def update_teacher_params(student, teacher, alpha, global_step):
    # Use the true average until the exponential average is more correct
    alpha = min(1 - 1 / (global_step + 1), alpha)
    for ema_param, param in zip(teacher.parameters(), student.parameters()):
        ema_param.data.mul_(alpha).add_(1 - alpha, param.data)
```

```
# backward
optimizer.zero_grad()
loss.backward()
optimizer.step()
global_step += 1
update_teacher_params(student, teacher, 0.995, global_step)
```