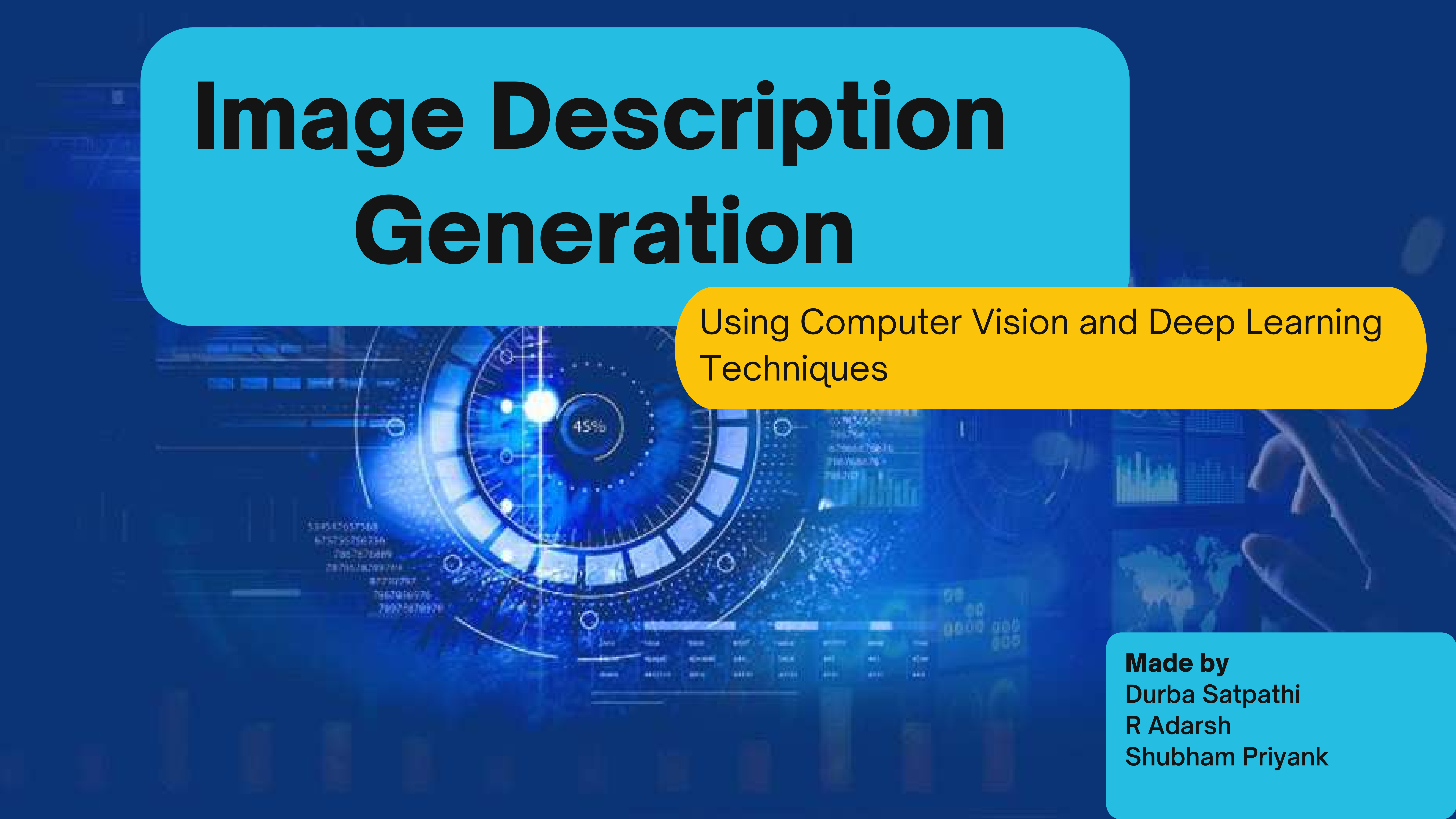
Image Description Generation

Using Computer Vision and Deep Learning Techniques



Made by
Durba Satpathi
R Adarsh
Shubham Priyank

Contents

Introduction

Date of Start: 24th June 2021
Project Mentor: Mr. Vivek Ananthan
Business Unit: COE Analytics
Title of the Project: " *Image Description Generation using Computer Vision and Deep Learning*"
Project Areas: Computer Vision, Deep Learning, Natural Language Processing
About the Project: Image description generation or image captioning is the process of generating sentences related to a given image. Image description generation can be considered an end to end seq2seq problem because it deals with the conversion of images (sequence of pixels) to sentences (sequence of words).
Technology Used-We decided to use Google Colab for training our model. We worked with python3 and used many of its libraries primarily Pytorch, CV2, PIL, Pandas and Pickle.

Dataset Preparation

Caption Preprocessing

- Removing punctuation
- Converting to lower case
- Adding <start>, <end> and <pad> tokens
- Replacing every word of the sentence with its index number

Image Preprocessing

- Resizing the image
- Normalizing the image
- Scaling the image

Image Augmentaion

- Center crop
- Brightening
- Adding Blur

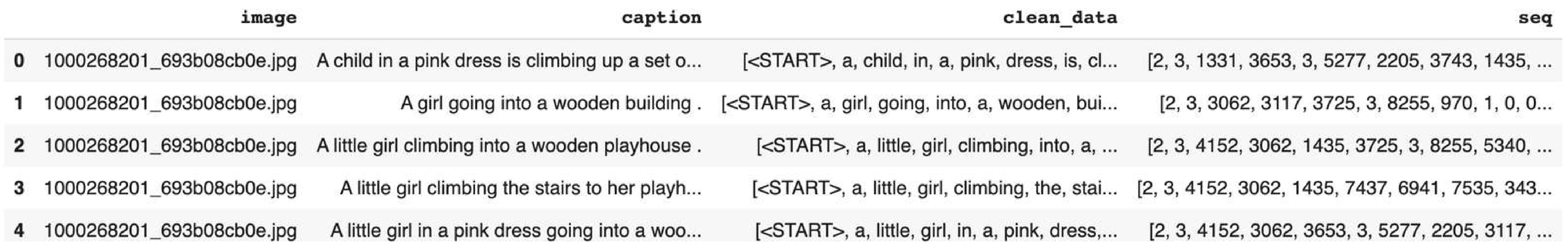


	image	caption	clean_data	seq
0	1000268201_693b08cb0e.jpg	A child in a pink dress is climbing up a set o...	[<START>, a, child, in, a, pink, dress, is, cl...	[2, 3, 1331, 3653, 3, 5277, 2205, 3743, 1435, ...
1	1000268201_693b08cb0e.jpg	A girl going into a wooden building .	[<START>, a, girl, going, into, a, wooden, bui...	[2, 3, 3062, 3117, 3725, 3, 8255, 970, 1, 0, 0...
2	1000268201_693b08cb0e.jpg	A little girl climbing into a wooden playhouse .	[<START>, a, little, girl, climbing, into, a, ...	[2, 3, 4152, 3062, 1435, 3725, 3, 8255, 5340, ...
3	1000268201_693b08cb0e.jpg	A little girl climbing the stairs to her playh...	[<START>, a, little, girl, climbing, the, stai...	[2, 3, 4152, 3062, 1435, 7437, 6941, 7535, 343...
4	1000268201_693b08cb0e.jpg	A little girl in a pink dress going into a woo...	[<START>, a, little, girl, in, a, pink, dress,...	[2, 3, 4152, 3062, 3653, 3, 5277, 2205, 3117, ...

Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset.
Training deep learning neural network models on more data can result in more skillful models, and the augmentation techniques can create variations of the images that can improve the ability of the fit models to generalize what they have learned to new images.

Image Augmentation- Center crop

Performed centre crop by taking a 299 by 299-pixel frame at the centre of each image and cropping it.

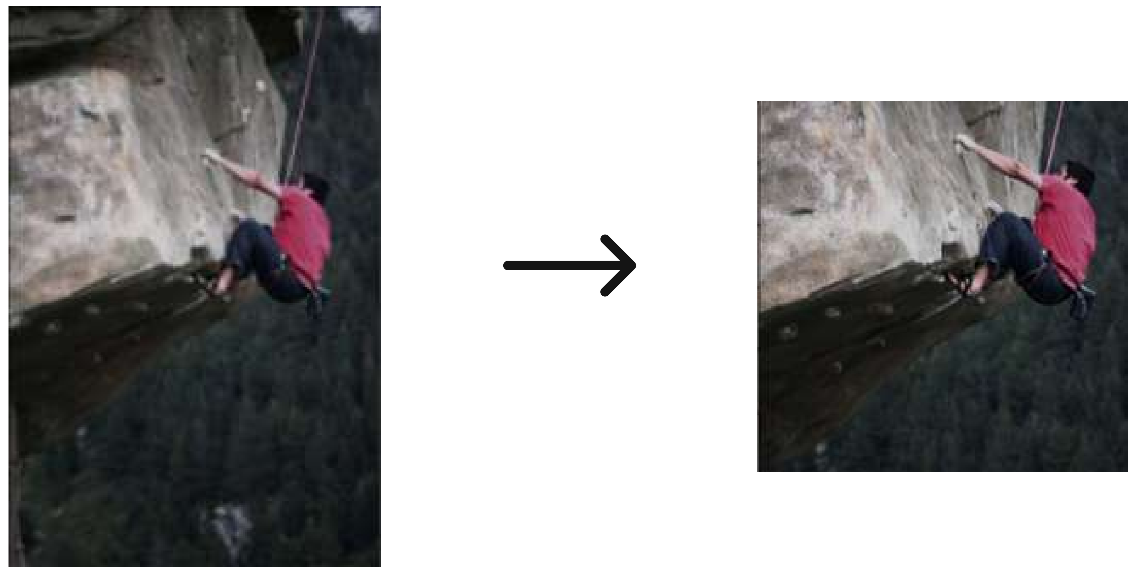


Image Augmentation- Brightening

Increased brightness of the image by adding a random value to the colour value channel of the image.

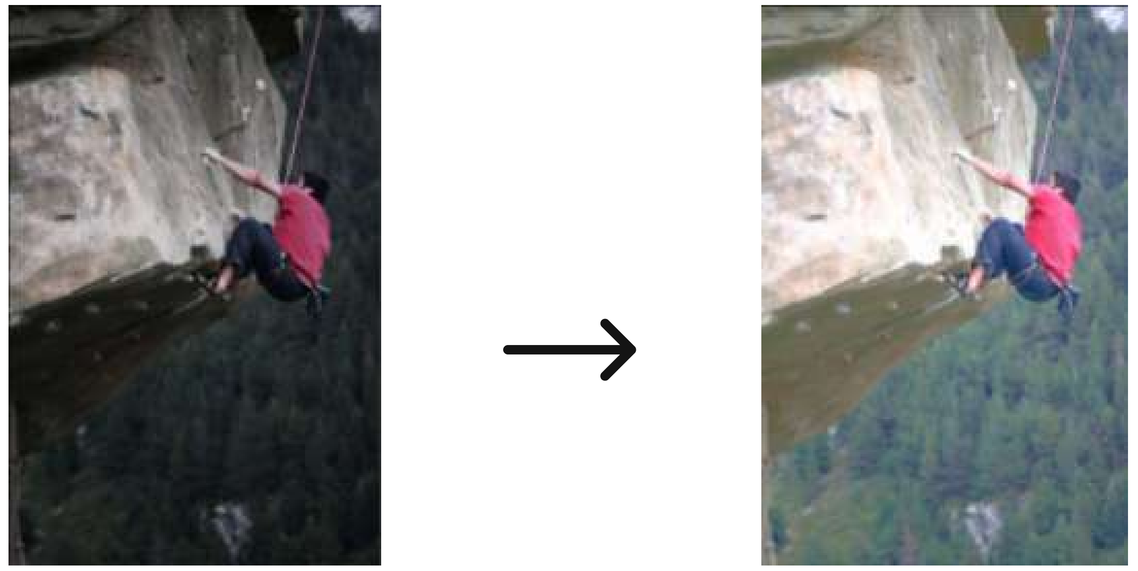
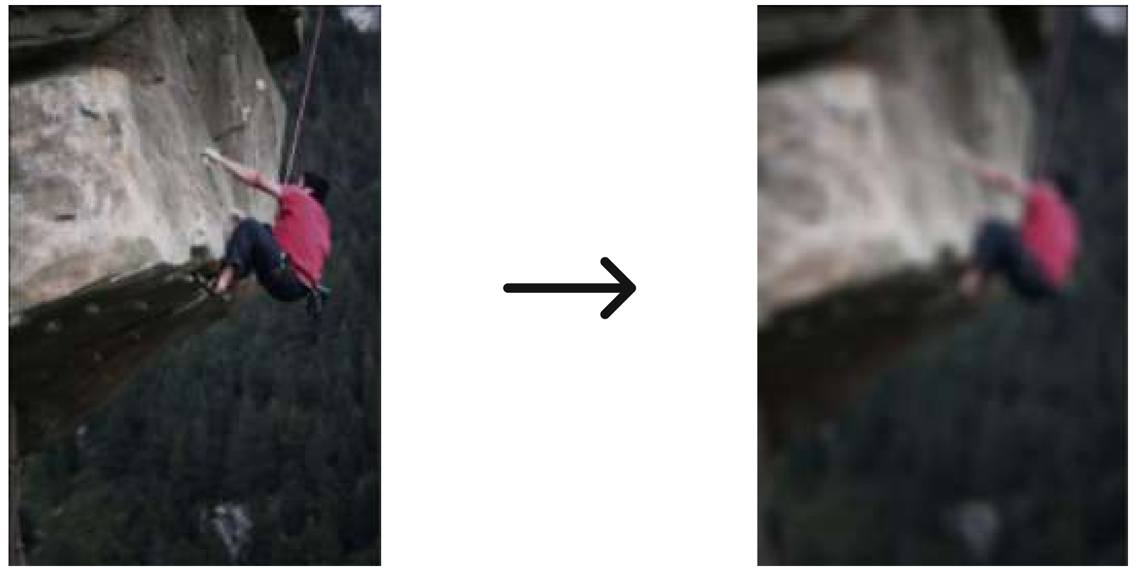


Image Augmentation- Blurring

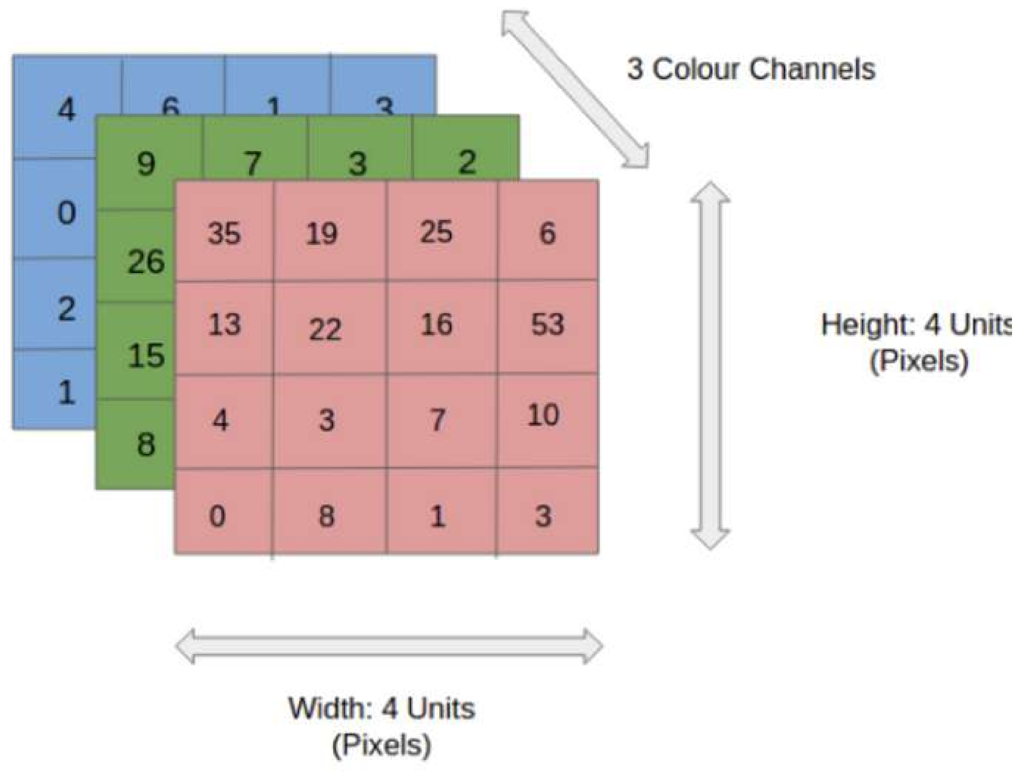
Adds blur to the image by using the blur function of the OpenCV library.



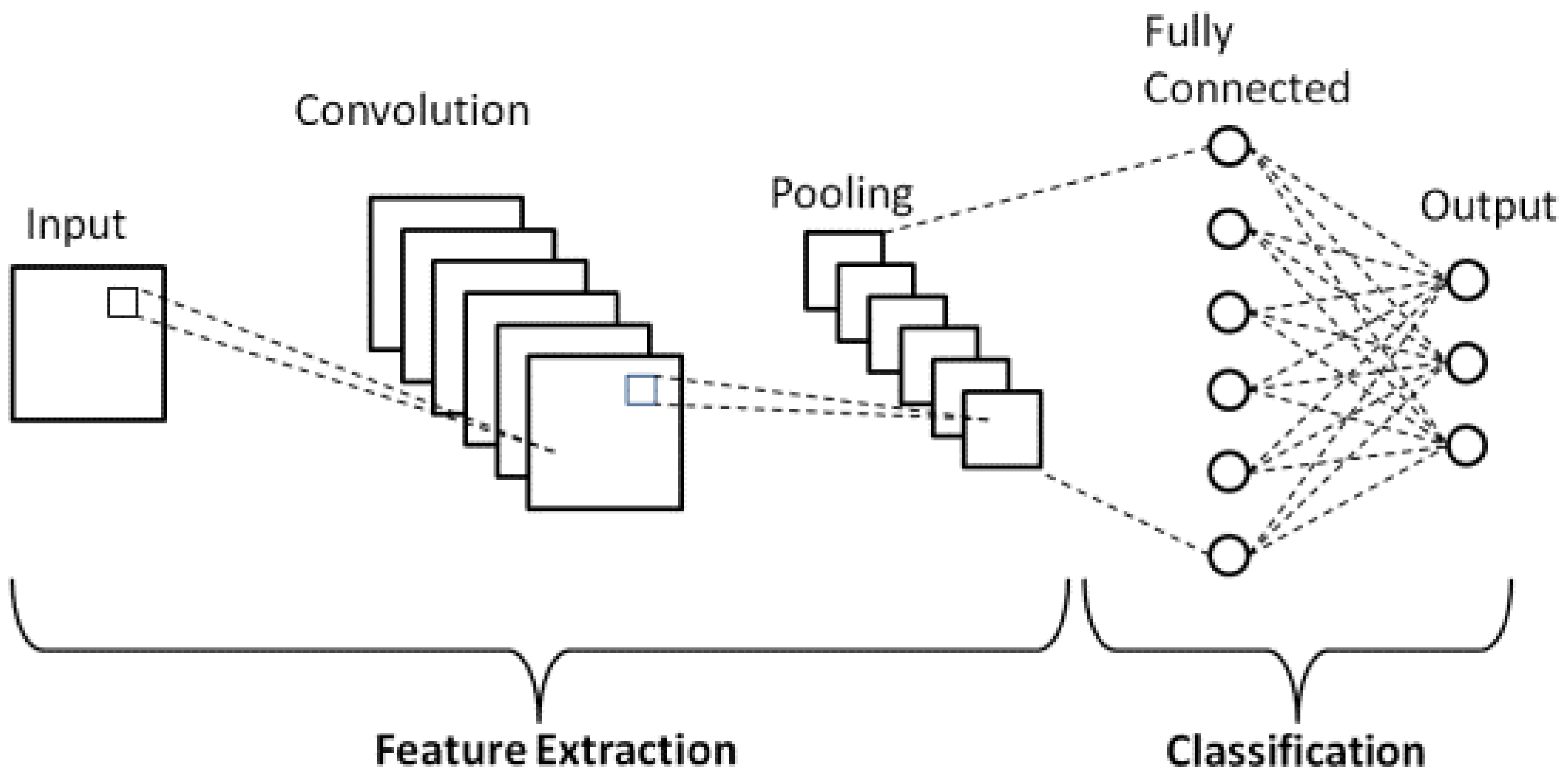
CNNs

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.

Visual Cortex



4×4×3 RGB Image



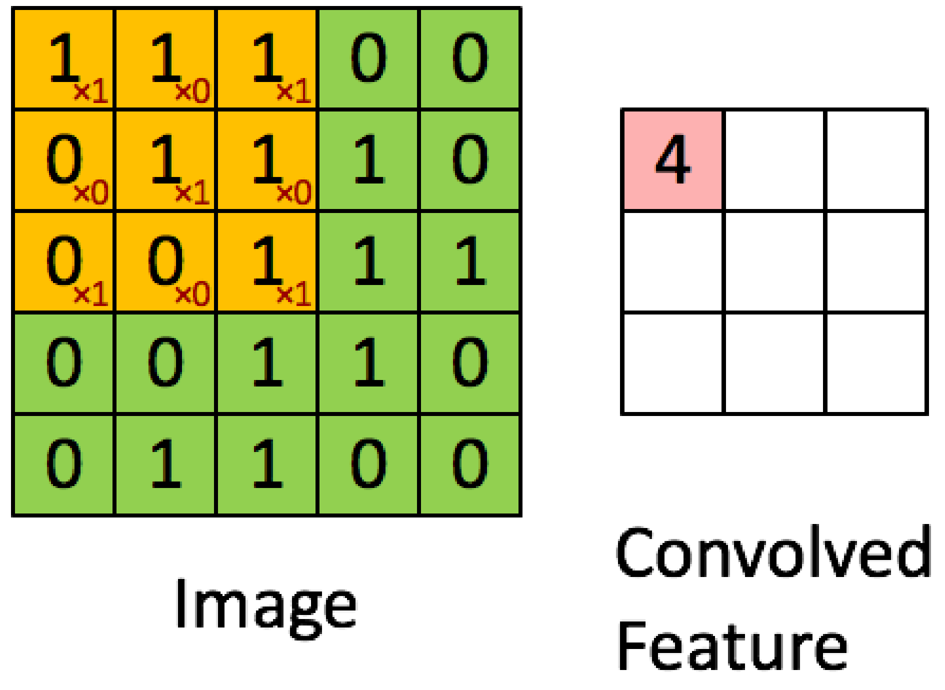
Convolution Layer

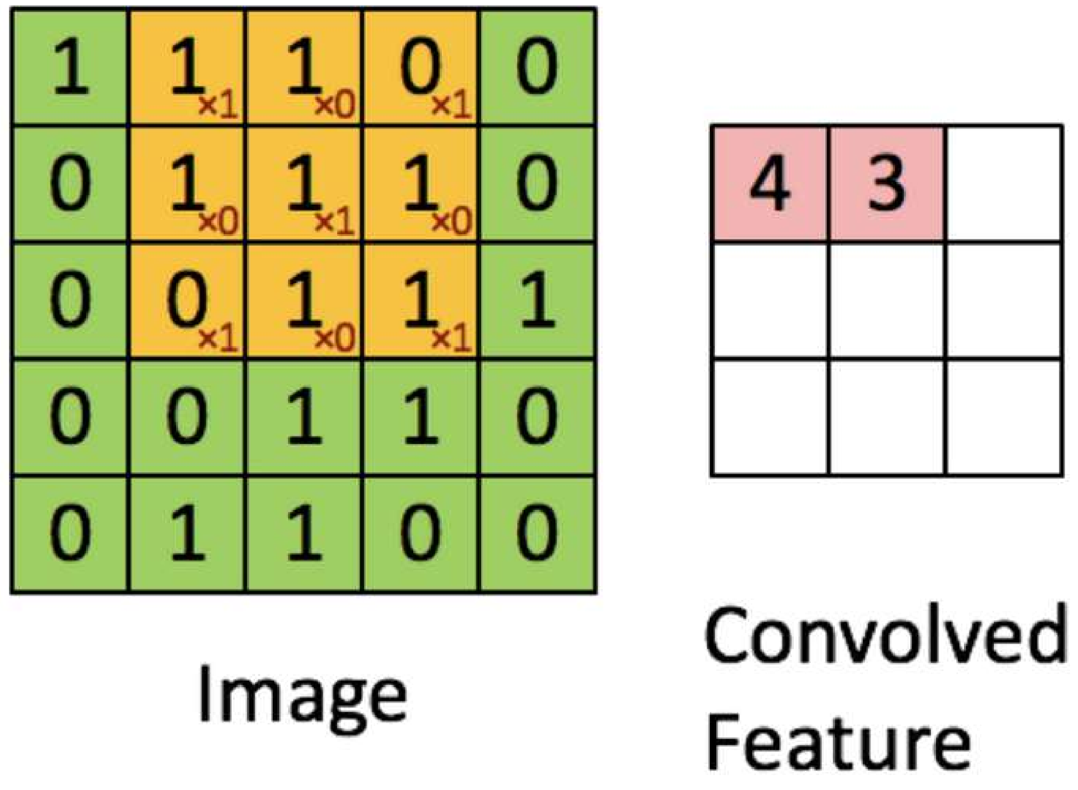
- The objective of the Convolution Operation is to extract the high-level features such as edges, from the input image.

$$s(t) = (x^*w)(t) = \int x(a)w(t\text{-}a)da$$

$$s(t) = (x^*w)(t) = \sum_{-\propto}^{\propto} x(a)w(t\text{-}a)$$

$$s(i,j) = (I^*K)(i,j) = \sum_{m}\sum_{n} I(m,n)K(i\text{-}m,j\text{-}n)$$





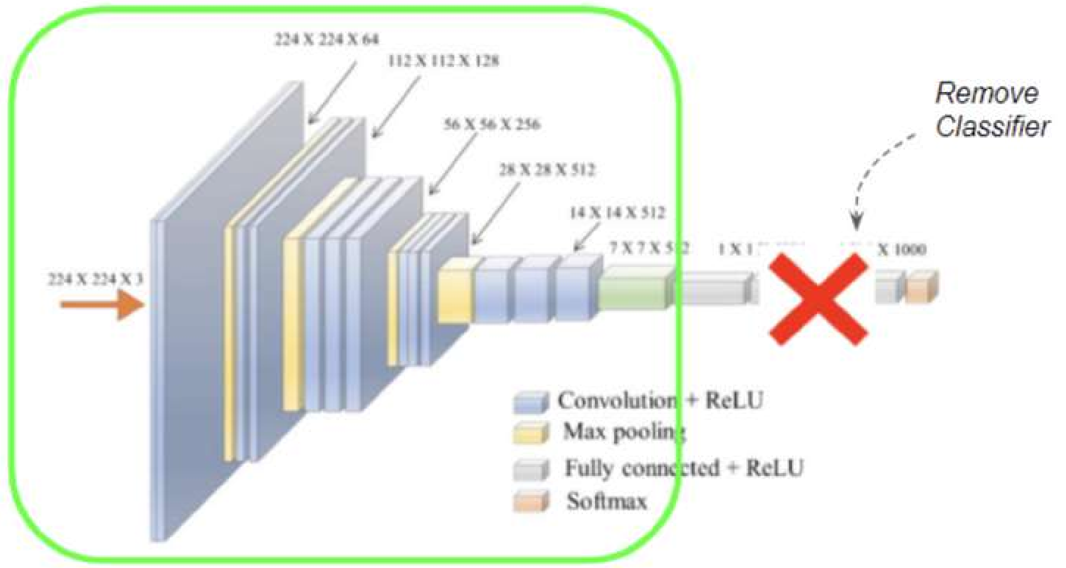
Pooling Layer

- The Pooling layer is responsible for reducing the spatial size of the Convolved Feature.
- Max Pooling and average pooling

- **Transfer learning** involves using models trained on one problem as a starting point on a related problem.

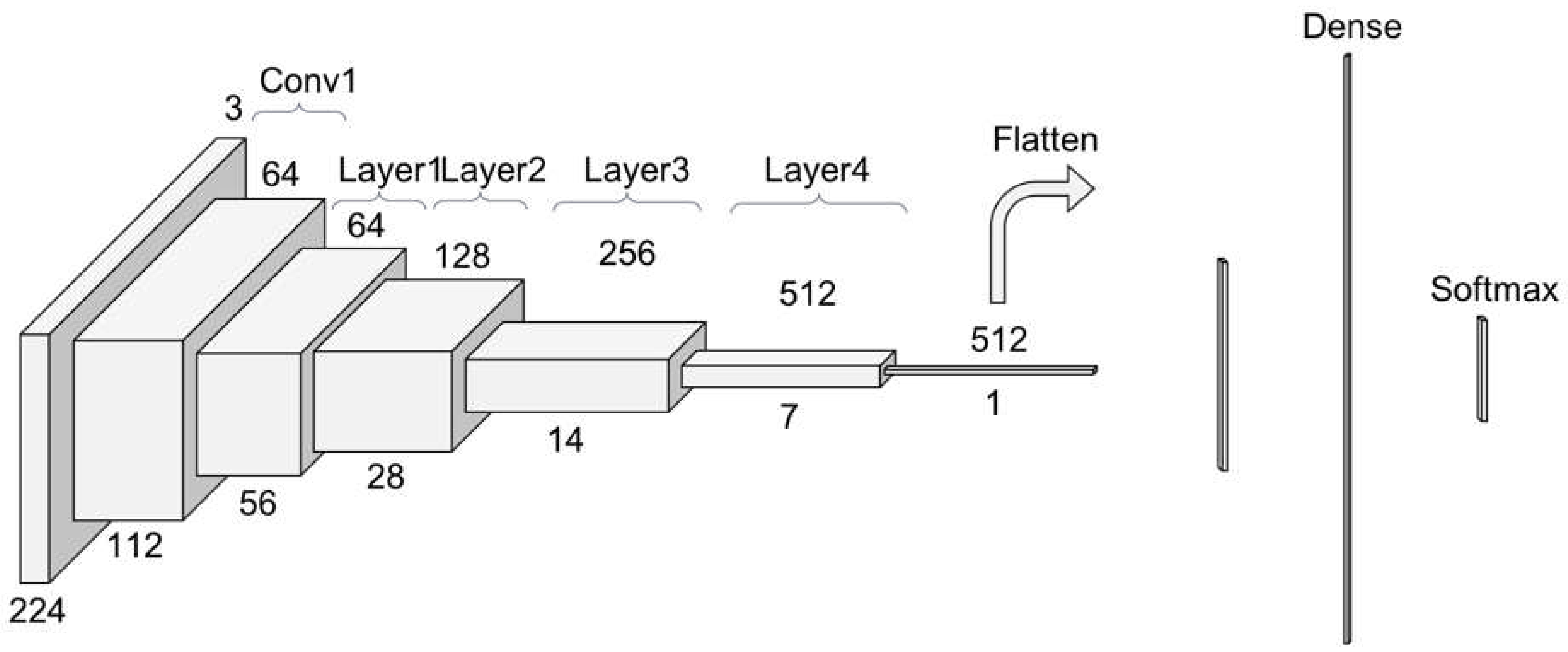
Fully Connected Layer

- Now that we have converted our input image into a suitable form for our Multi-Level Perceptron, we shall flatten the image into a column vector. The flattened output is fed to a feed-forward neural network and backpropagation applied to every iteration of training.

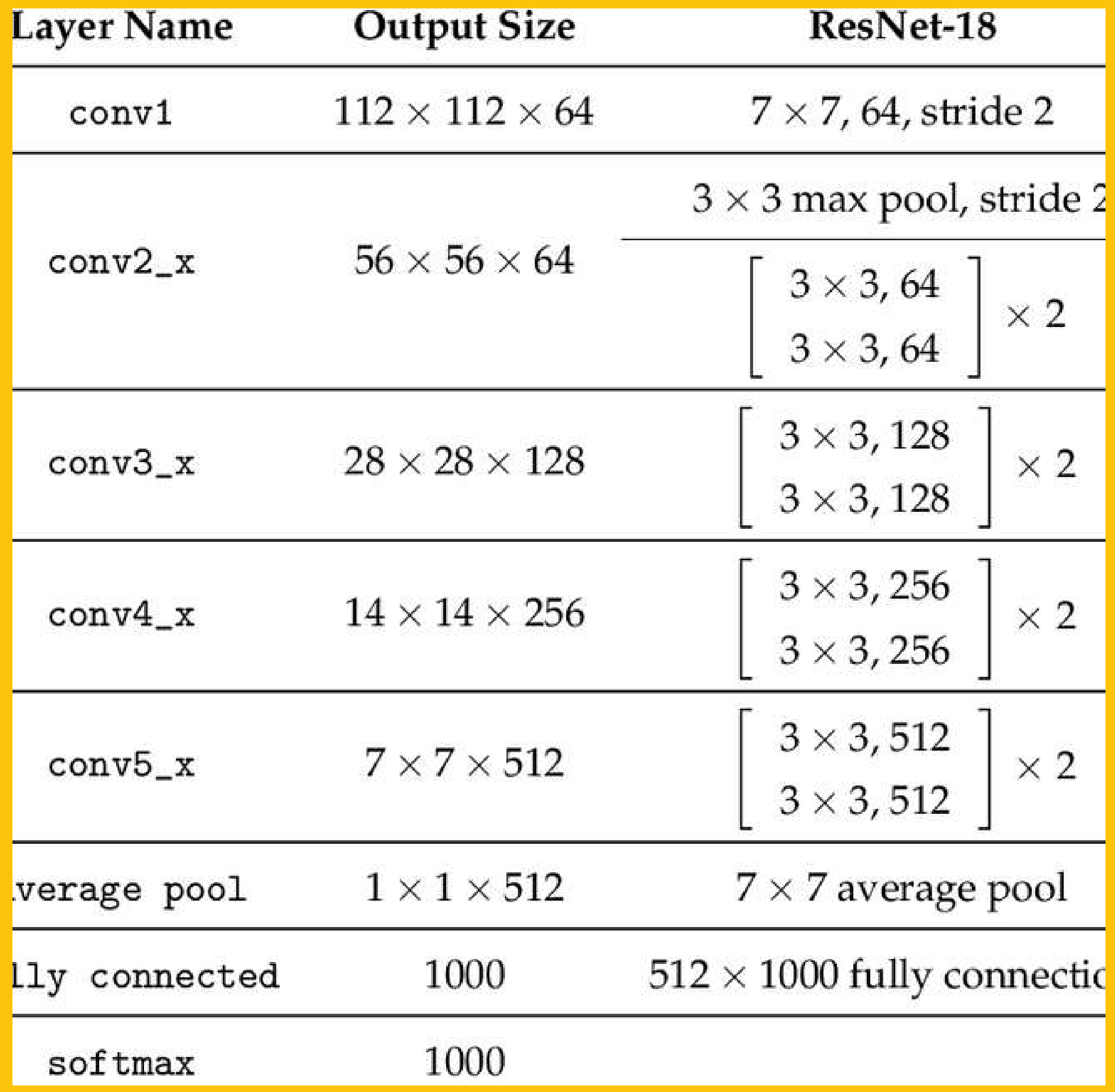


Encoder Network-1 ResNet18

- Used as encoder CNN to extract the features of images and get the vector embeddings.
- The images were resized as 224*224 (the input shape of ResNet18)
- The images were then normalized and converted to tensors.
- Created a data loader of transformed images
- Forward feed the transformed images through ResNet18
- Output embedding obtained from the fourth layer of the ResNet.
- Output embedding - shows 512 feature maps of dimension 7 X 7.



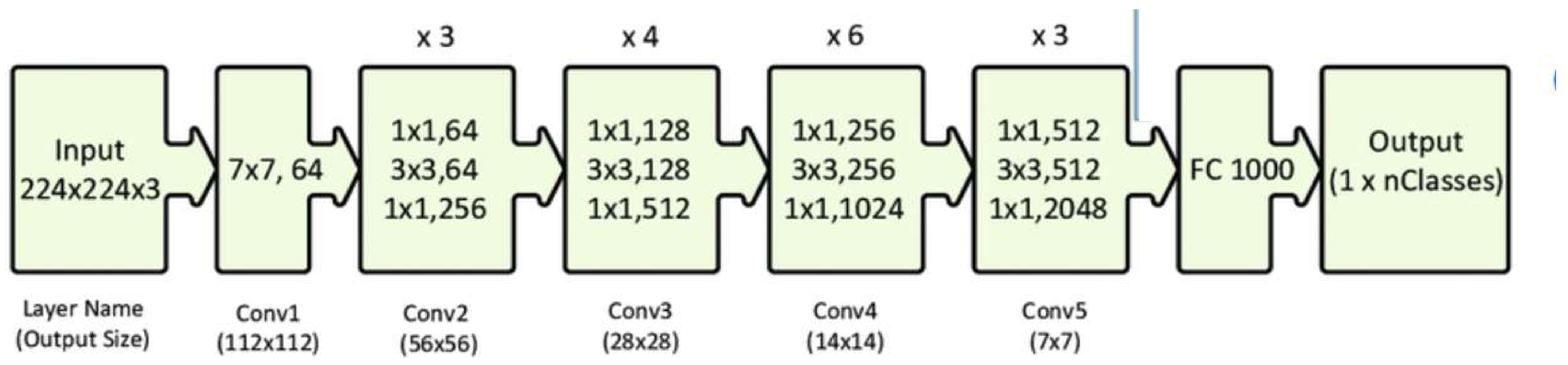
RESNET-18 Layers



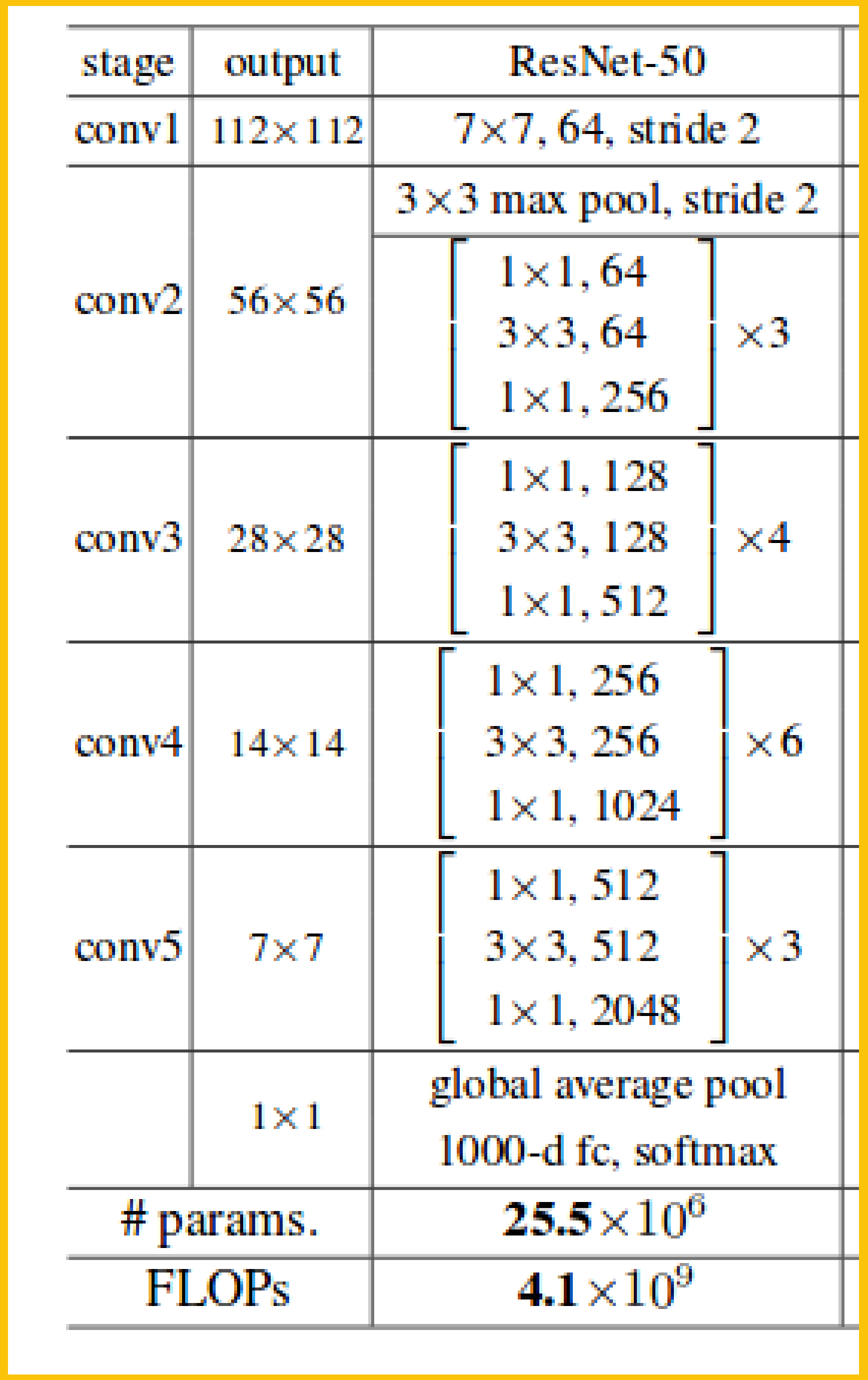
ayer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
verage pool	$1 \times 1 \times 512$	7×7 average pool
lly connected	1000	512×1000 fully connectic
softmax	1000	

Encoder Network-2 ResNet50

- Similar to ResNet18 the images are resized to 224*224 dimension (the input shape of the ResNet model).
- Then these images are normalized, converted to tensors, and passed through ResNet50.
- The output is then obtained from the fourth layer of the model which is a 2048 feature map of dimension 7×7.



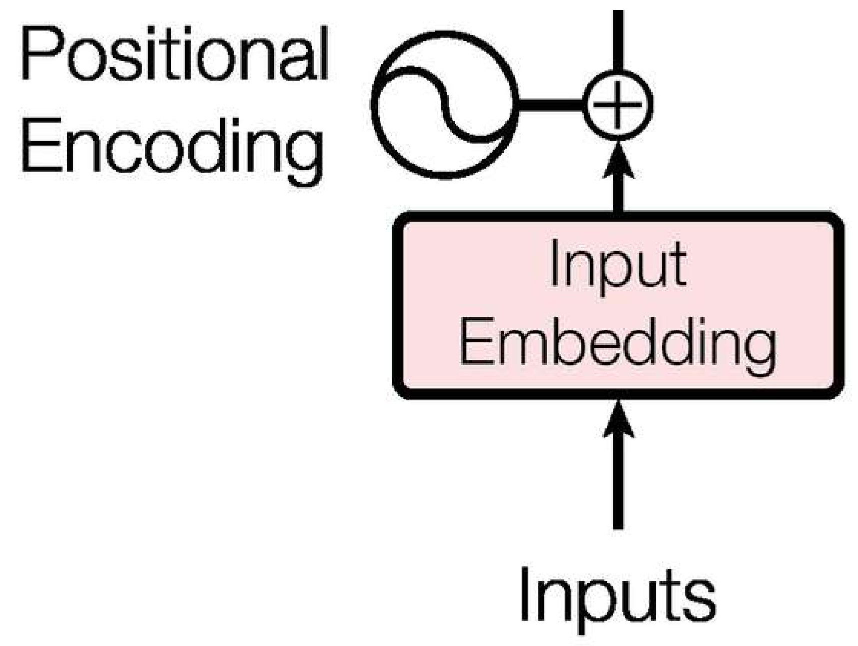
RESNET-50 Layers

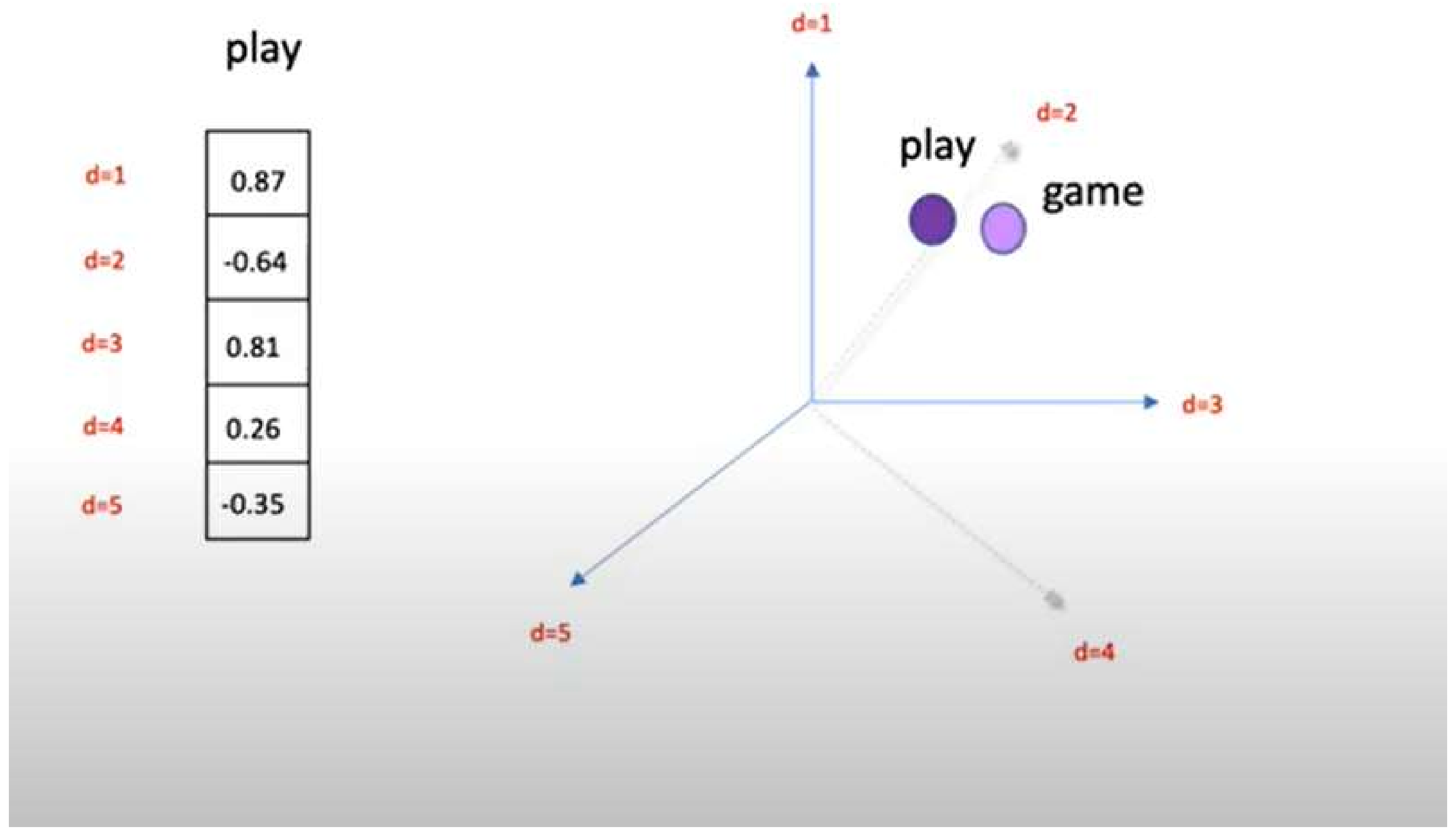


stage	output	ResNet-50
conv1	112×112	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2
		$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3	28×28	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$
conv4	14×14	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$
conv5	7×7	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1×1	global average pool 1000-d fc, softmax
# params.		**25.5**$\times10^6$
FLOPs		**4.1**$\times10^9$

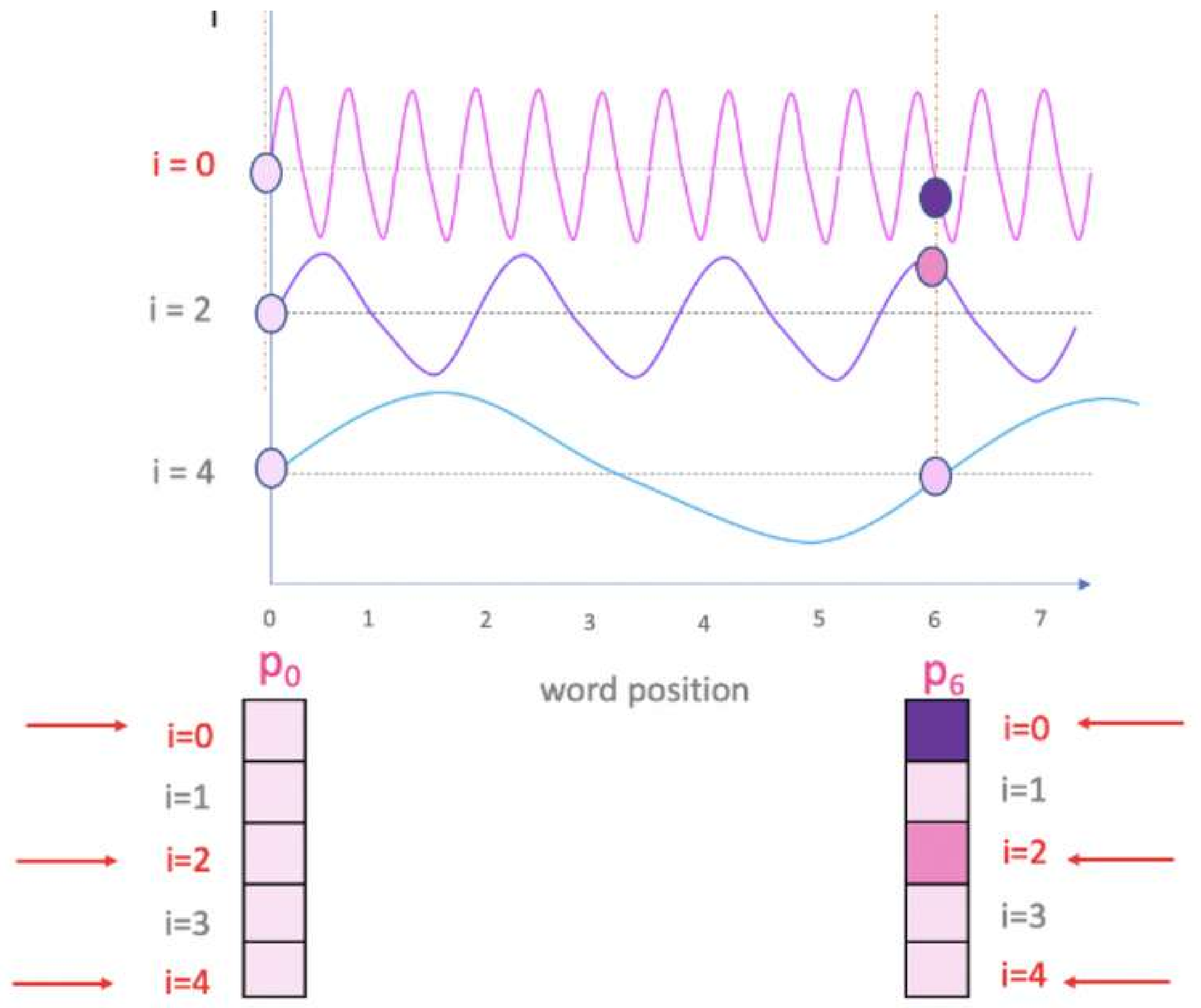
Preparing Caption Data for the Transformer Network

Input Embeddings: Word embeddings are generated that capture the linguistic features of the word.
Positional Encodings: Since every word flows into the decoder stack of the transformer simultaneously, we used positional encodings to incorporate the position of the word in the sentence to the model.





$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$



Decoder Network

TRANSFORMER ARCHITECTURE

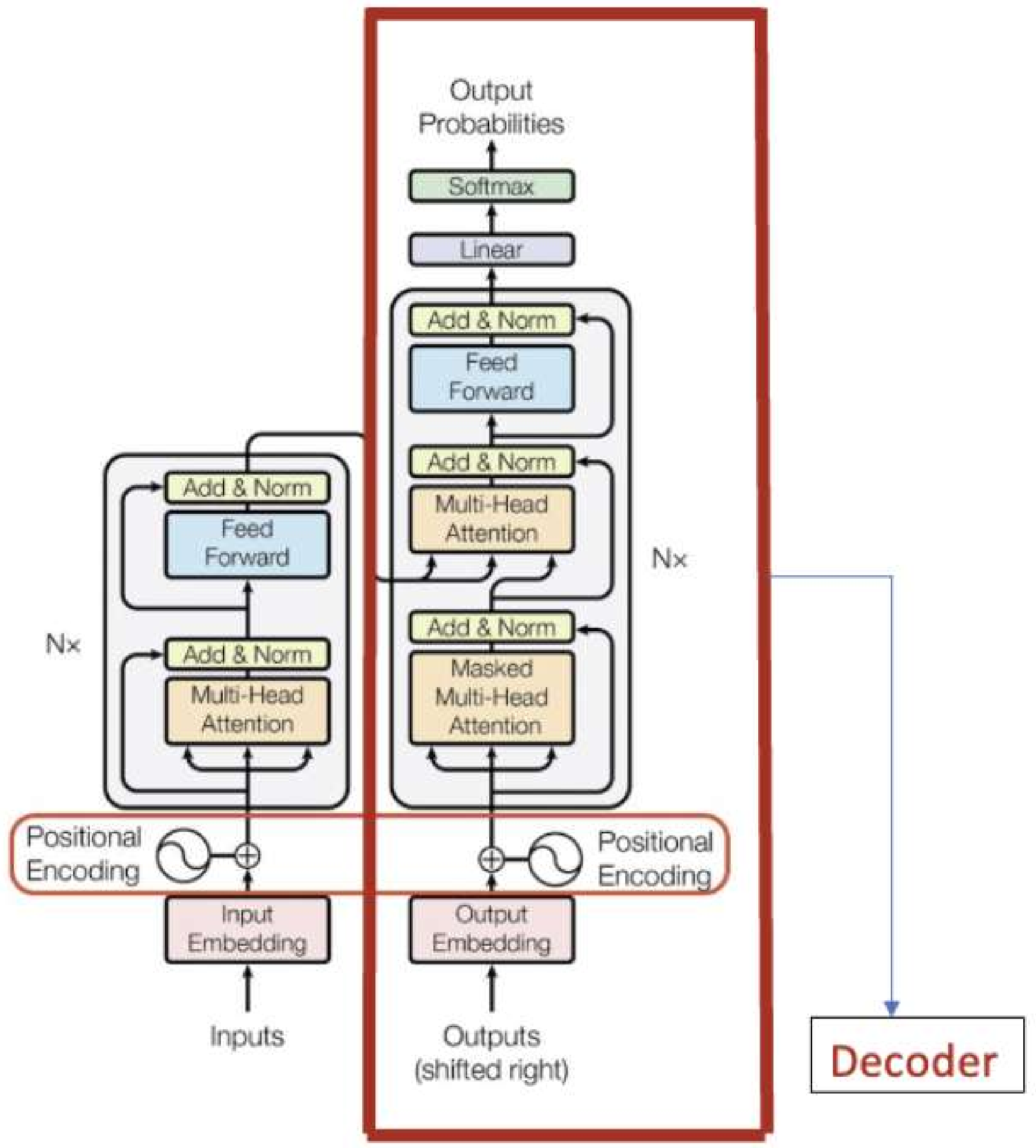


Fig-3.2.1 Architecture of Transformer

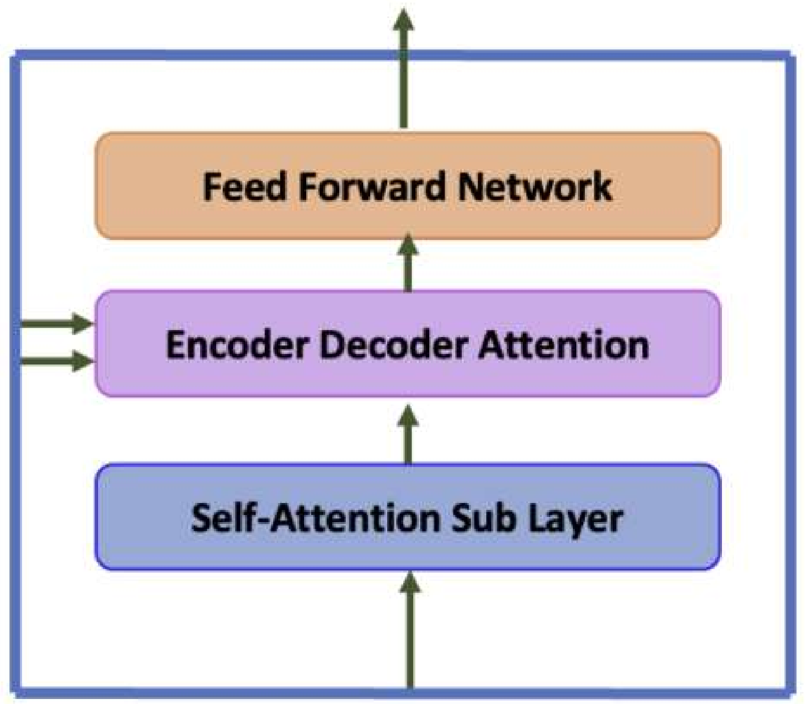
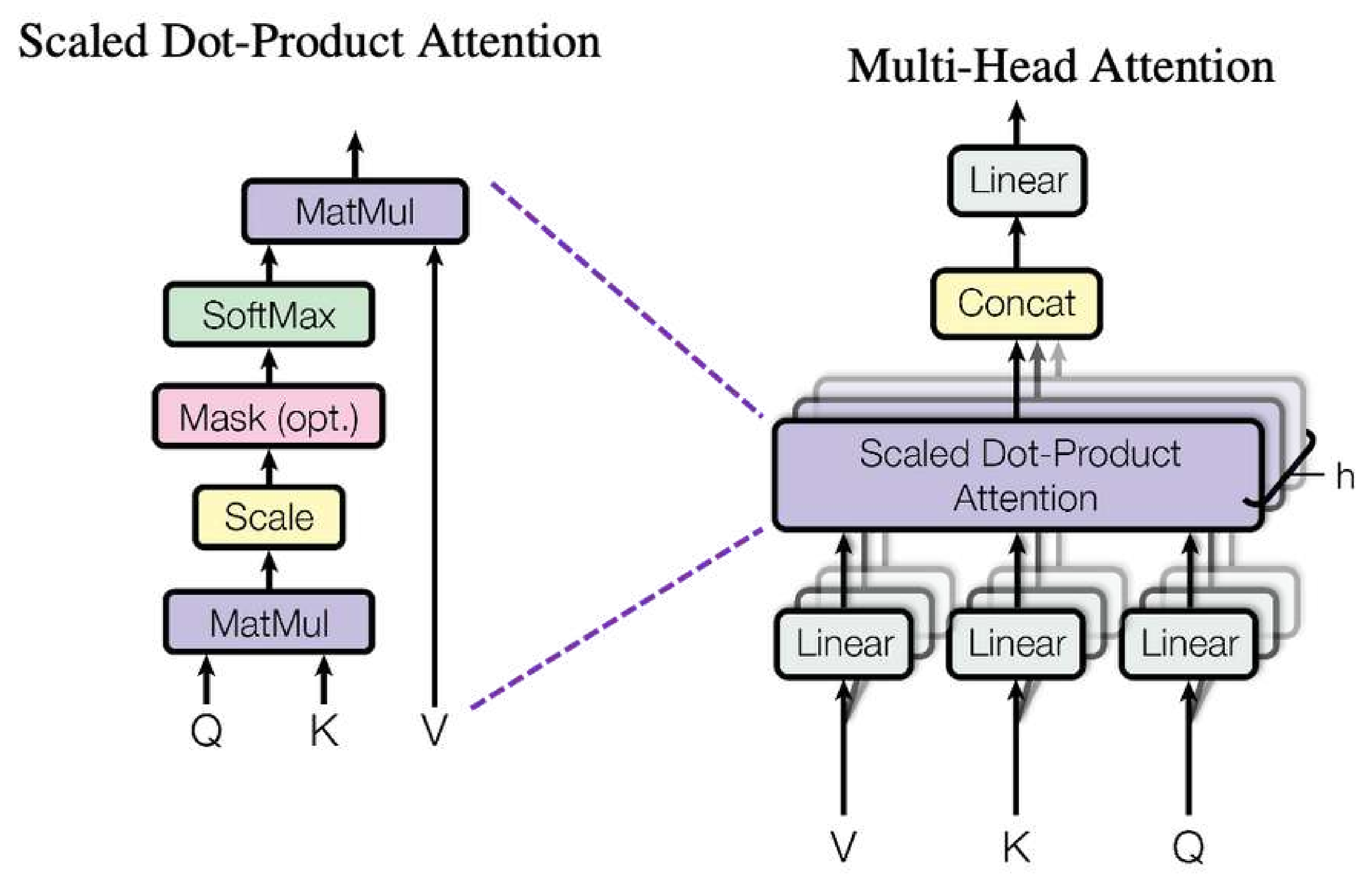


Fig-3.2.2 Sublayers in each Stack of a decoder

Image Source-[Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin; *Attention is all you need*,2017]

Attention

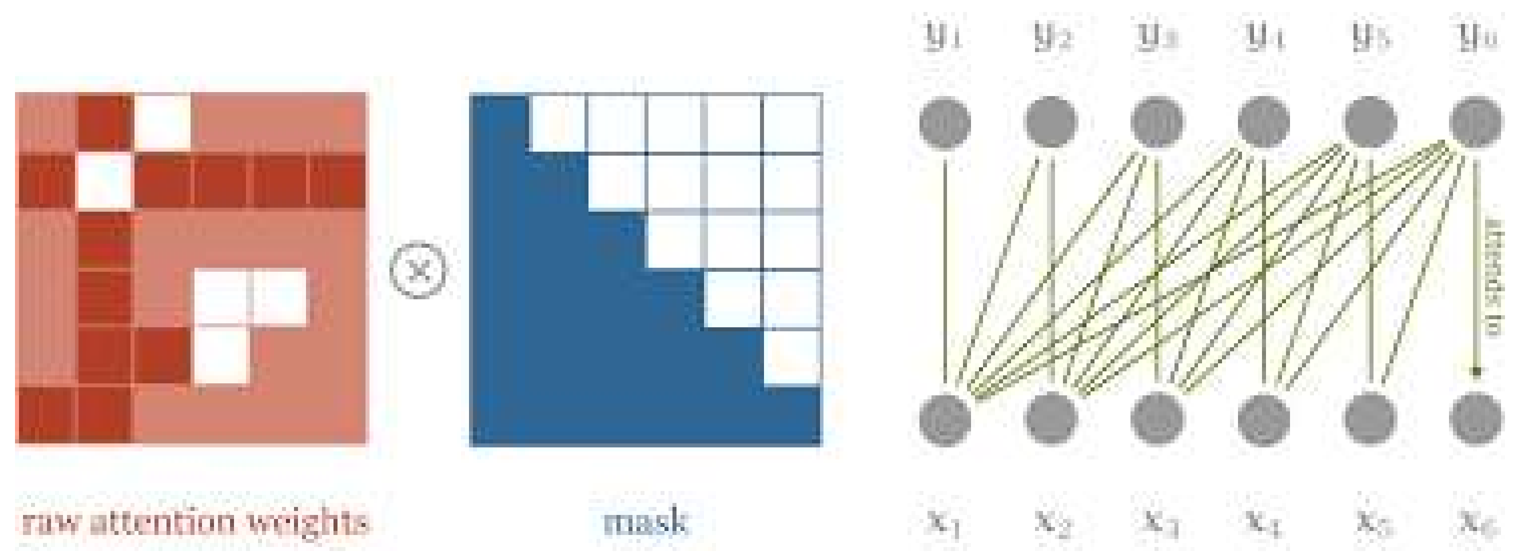
- Attention layer takes input as Query , Key and Value.
- **Multi-head Attention:** The attention module repeats the calculations multiple times in parallel. It splits the Q,K,V parameters N-ways and passes each split independently through a seperate head.
- This gives the transformer greater power to encode multiple relations & nuances of word



The scaled dot product attention is calculated as

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

$$MultiHeadAttention(Q,K,V) = Concat\,(head_1,\, head_2, \ldots .\, head_H)W^O$$

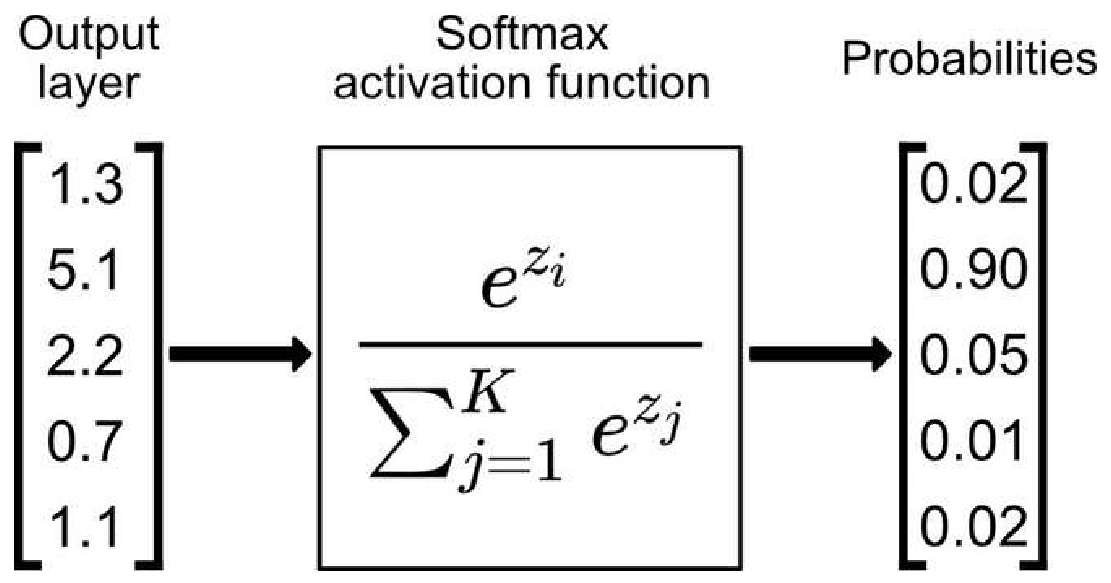

raw attention weights
mask
y_1 y_2 y_3 y_4 y_5 y_6
x_1 x_2 x_3 x_4 x_5 x_6
attends to

Masking

Masking is used to mask certain tokens that do not contribute to attention mechanism.

- **Padding Mask :** The zero-paddings at the end of sequence are not supposed to contribute to attention calculation/target sequence generation.
- **Look-Ahead Mask:** Only the words preceding the current word may contribute to the generation of the next word. Masked Multi-Head Attention ensures this.

$$\text{FeedForwardNetwork}(z)=\max(z\,W_1+b_1,0)\,W_2+b_2$$


Output layer
Softmax activation function
Probabilities
1.3
5.1
2.2
0.7
1.1
$\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$
0.02
0.90
0.05
0.01
0.02

Feed Forward Layer

- Above the attention layers there is a position wise fully connected simple feed forward network which consists of two linear transformations and a ReLu activation.

Training the model

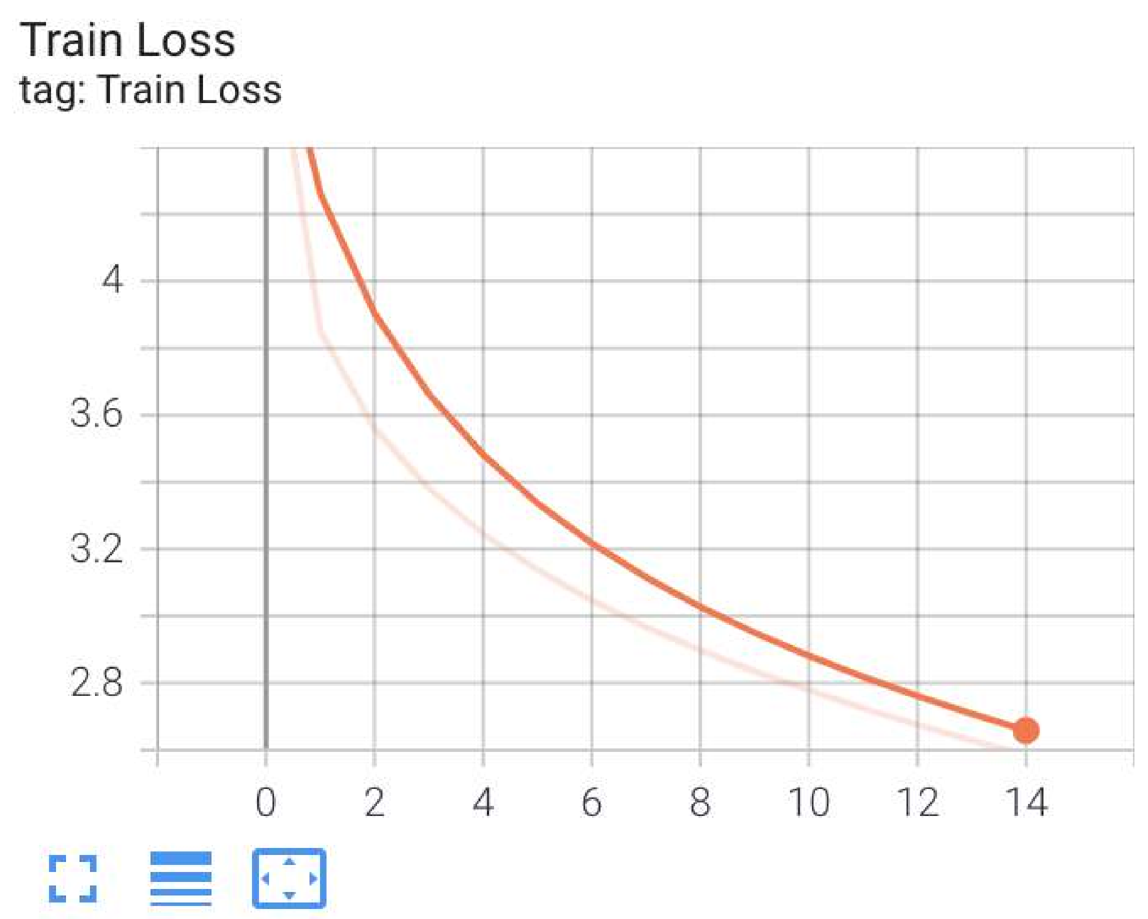
1.Hyperparameters
- Loss Metric- Cross Entropy Loss
- Optimizer-Adam

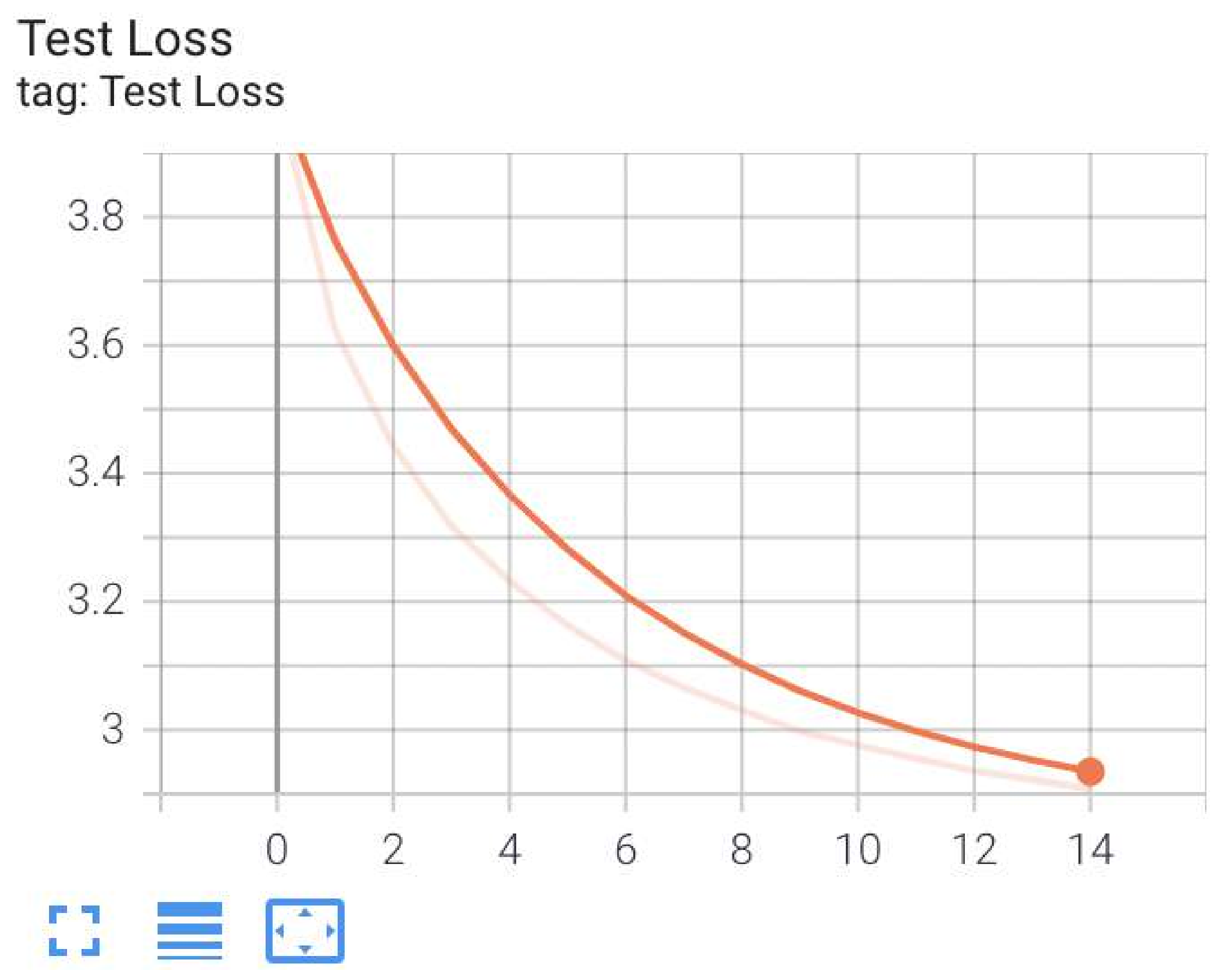
2.Reduce LR On Plateau

3.Used tesnsorboard to visualize the losses

Model (ResNet18+Transformers)

- Approximate Training Time: 30 minutes
- Initial Training Loss- 4.943099 Initial Testing Loss- 3.994659
- Final Training Loss- 2.58357 Final Testing Loss- 2.907768

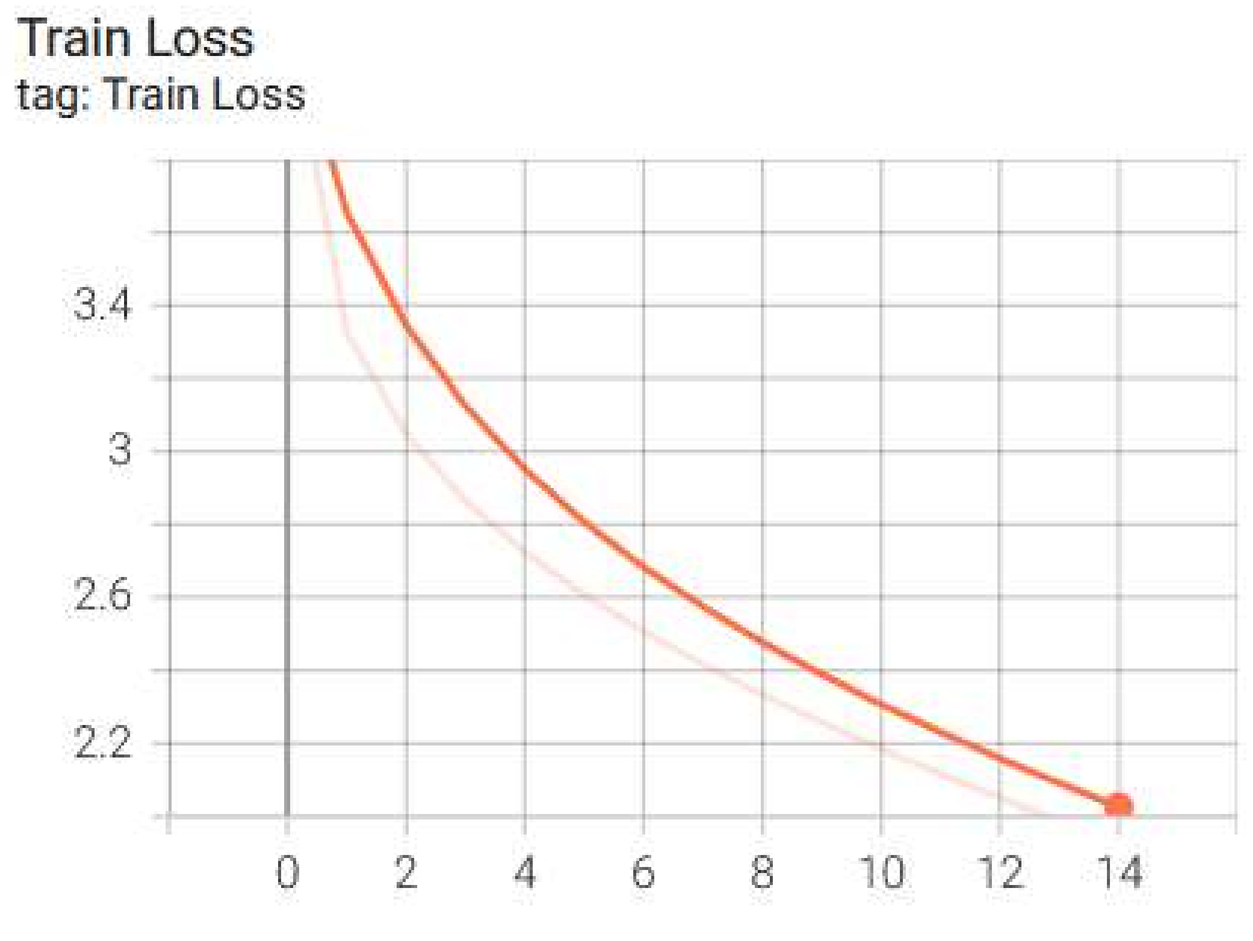


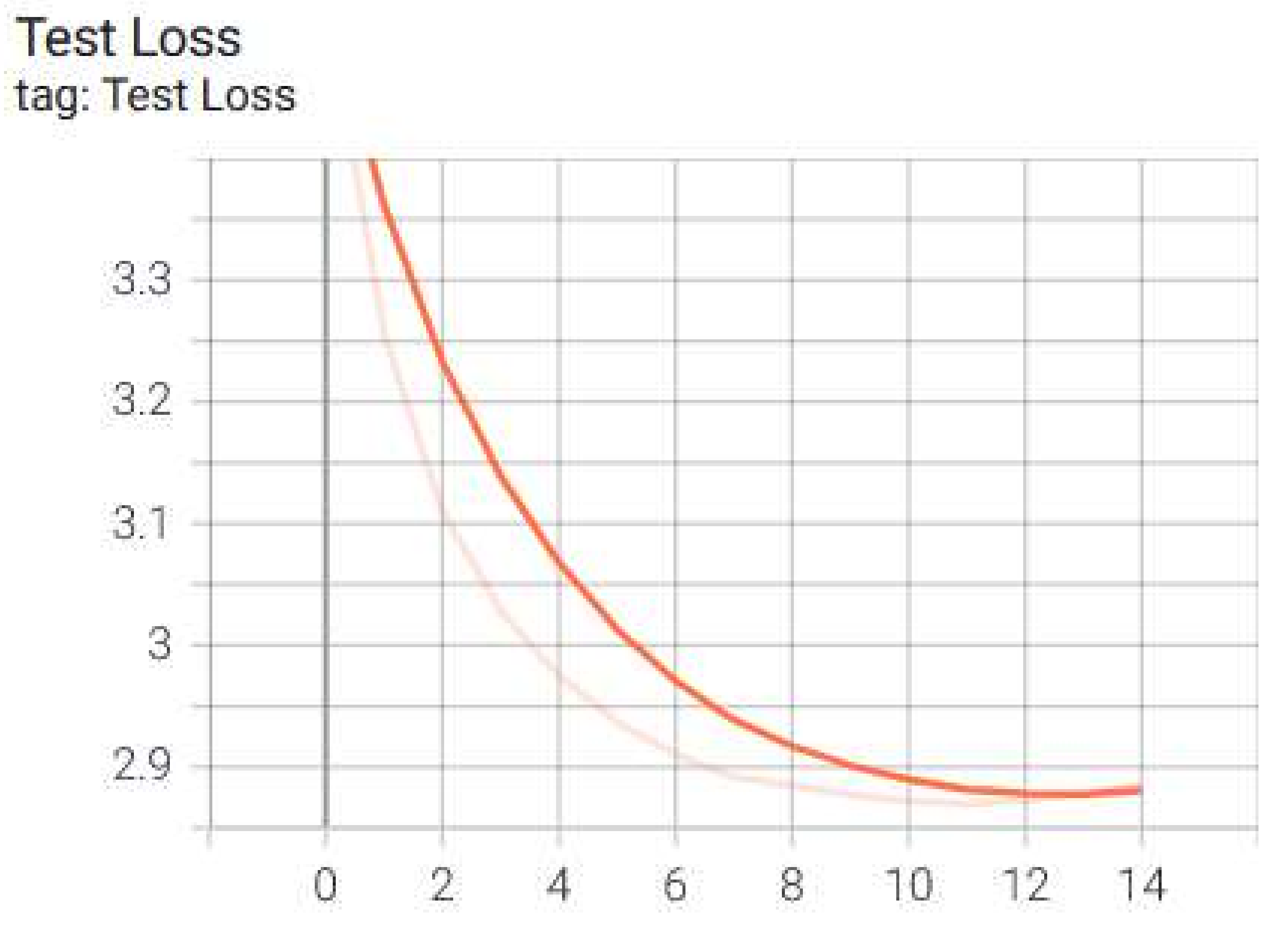


Hyperparameters
- Number of epochs-15
- Learning Rate-0.00001

Model (ResNet18+Transformers)

- With data augmentation
- Approximate Training Time: 50 minutes
- Initial Training Loss- 4.20113   Initial Testing Loss- 3.53871
- Final Training Loss- 1.92809   Final Testing Loss- 2.88611



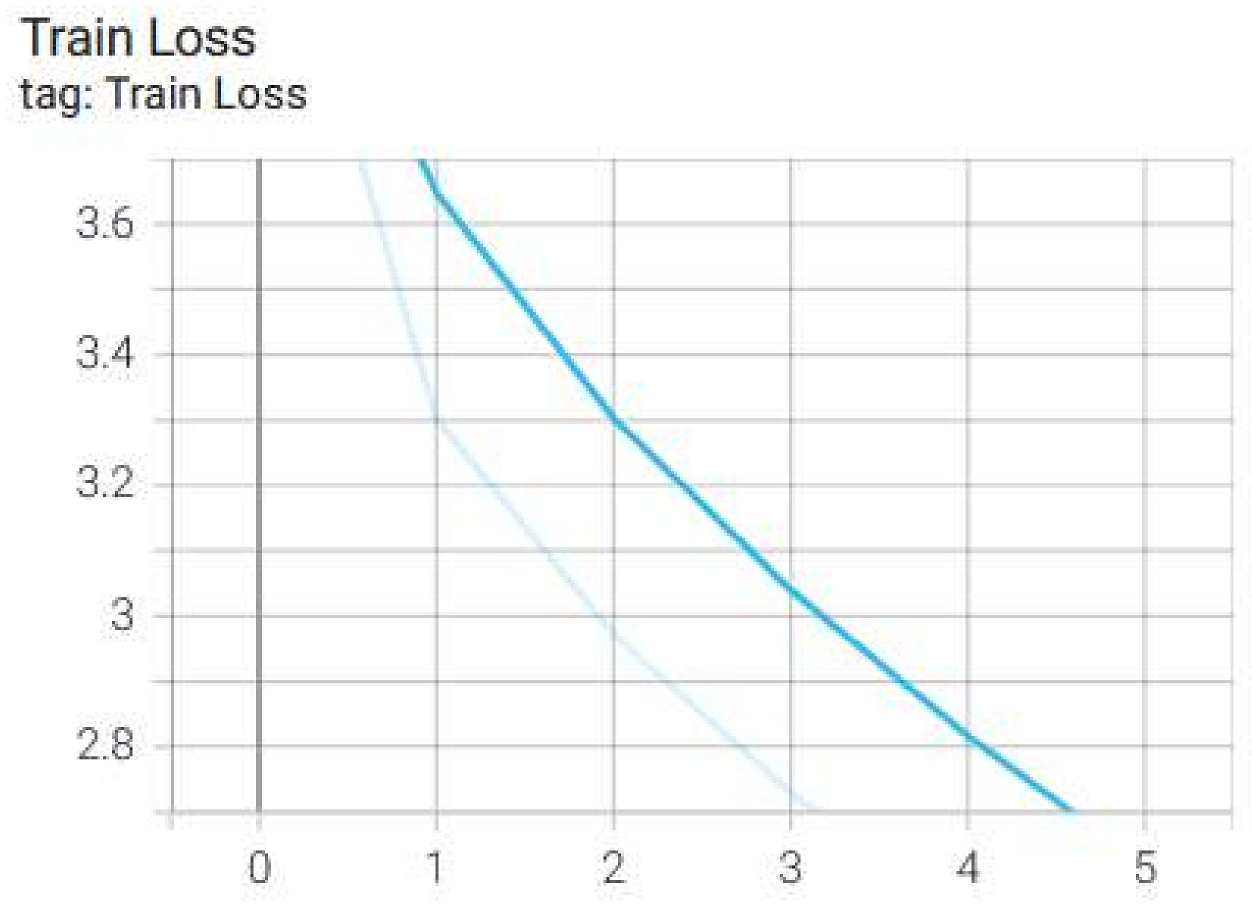


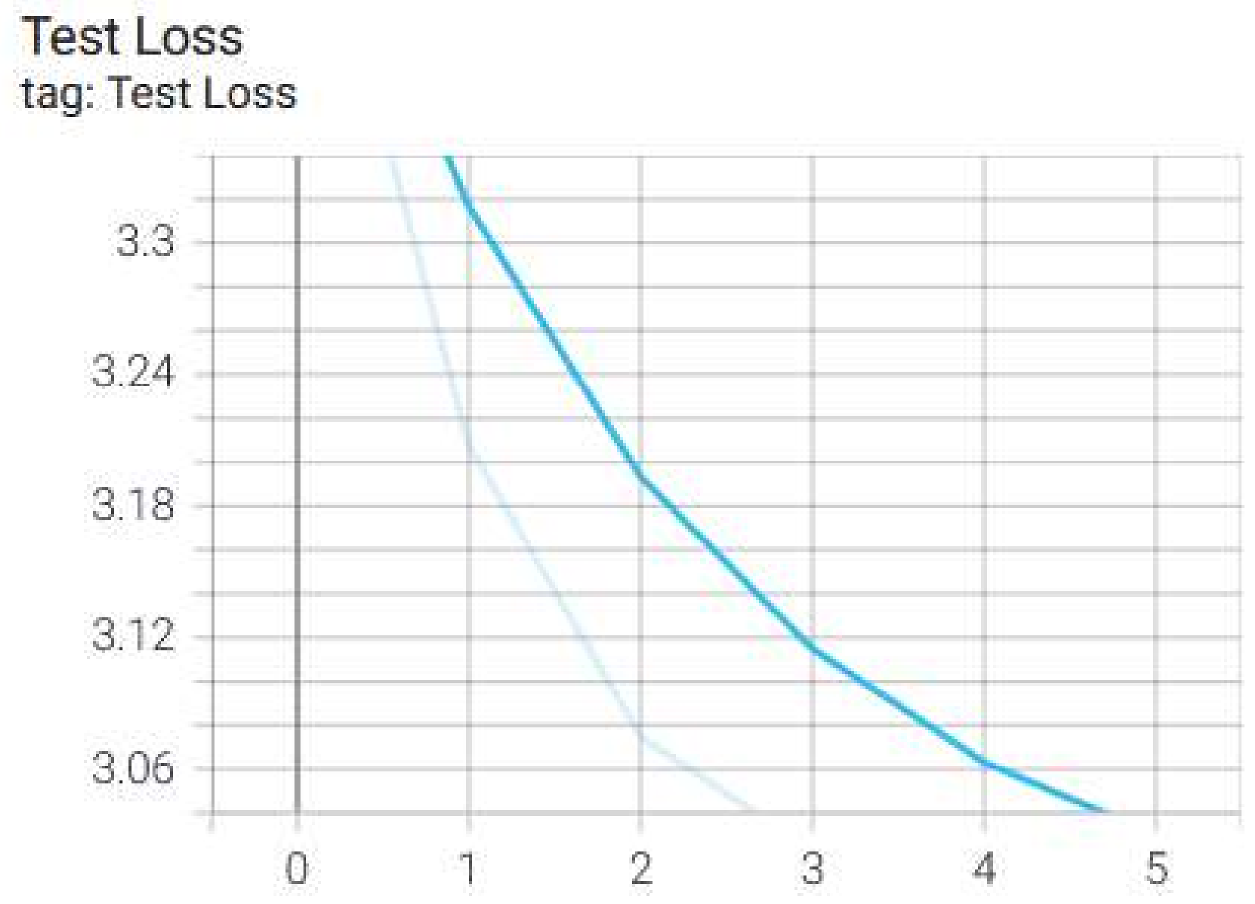
Hyperparameters

- Number of epochs-13
- Learning Rate- 0.00001

Model (ResNet50+Transformers)

- Approximate Training Time: 40 minutes
- Initial Training Loss- 4.22559   Initial Testing Loss- 3.49861
- Final Training Loss- 2.34179   Final Testing Loss- 2.98425



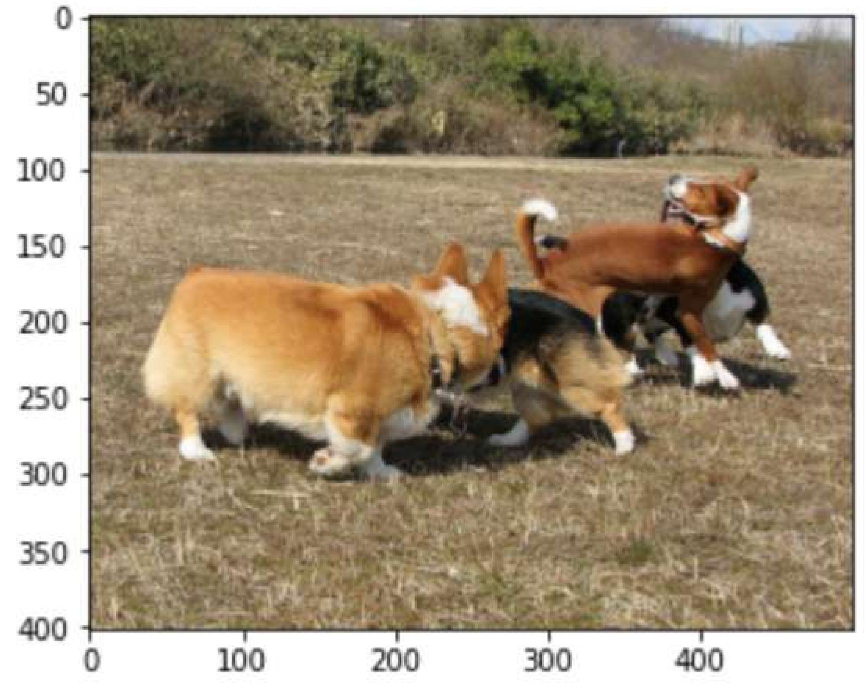


Hyperparameters
- Number of epochs-6
- Learning Rate- 0.00001

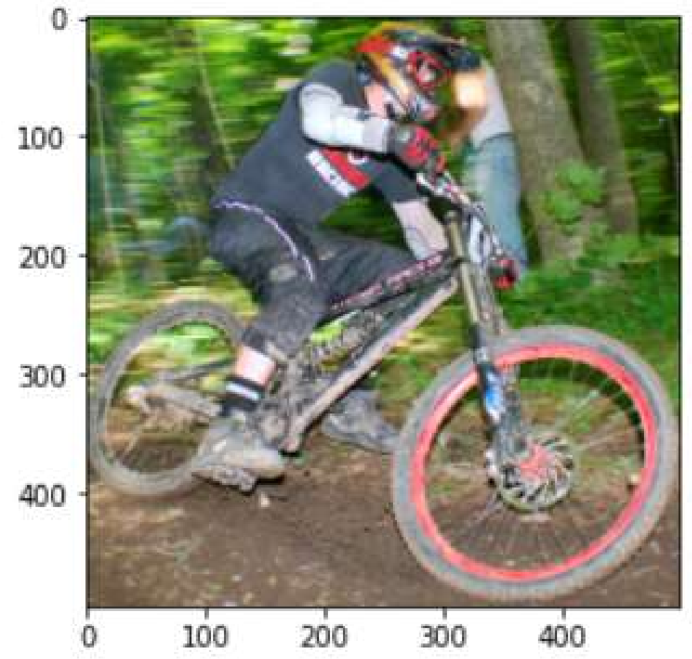
Generating captions of Test Set Images

```
Predicted caption :
two children are playing in a field .
```

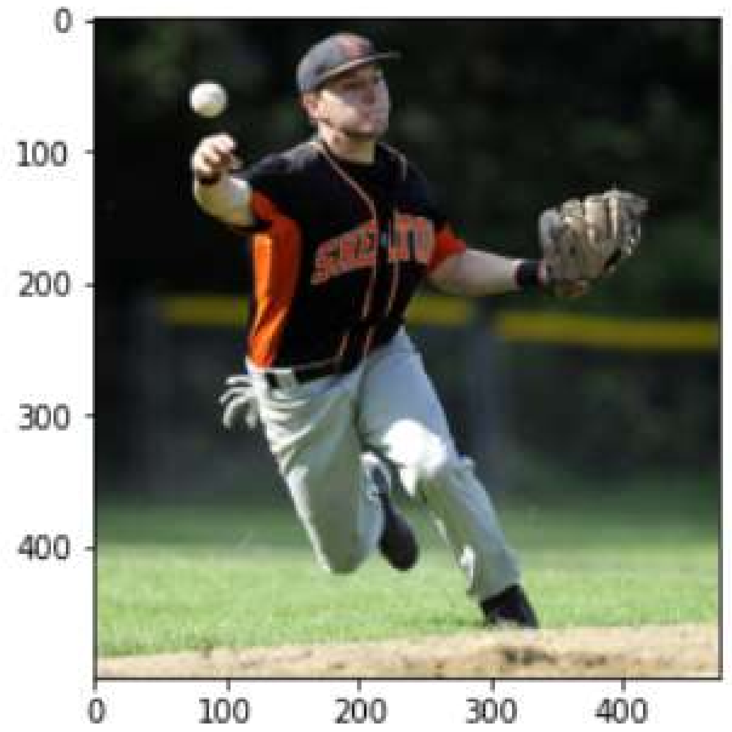
```
Predicted caption :
three dogs are playing in a field of grass .
```



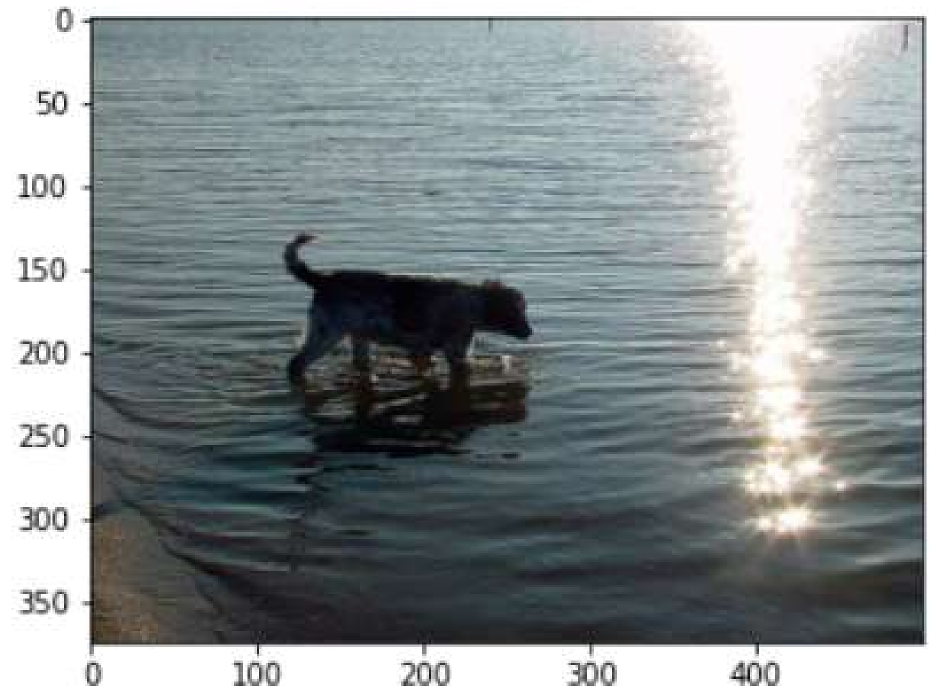
```
Predicted caption :
a man in a red helmet rides a bike through the woods .
```



```
Predicted caption :
a baseball player in a red uniform is throwing the ball .
```



```
Predicted caption :
a black dog is walking through the water .
```



Results

SNo	Encoder CNN	Decoder Network	Image Augmentation Done	Number of Epochs (Till which testing loss decreased)	Learning Rate	Final training Loss (Cross Entropy)	Final testing Loss (Cross Entropy)
1	ResNet18	Transformers	No	15	0.00001	2.583574	2.907768
2	ResNet18	Transformers	Yes	13	0.00001	1.988753	2.877248
3	ResNet50	Transformers	No	6	0.00001	2.341792	2.984258

Conclusion

- The best performing model among the models that we could train was that of ResNet18+Transformers along with image data augmentation.
- One of the most frequently occuring words in the vocabulary was dog. Thus the model worked quite well in identifying dogs in various backgrounds, even for images outside the test set
- The model sometimes could not distinguish between man and woman.

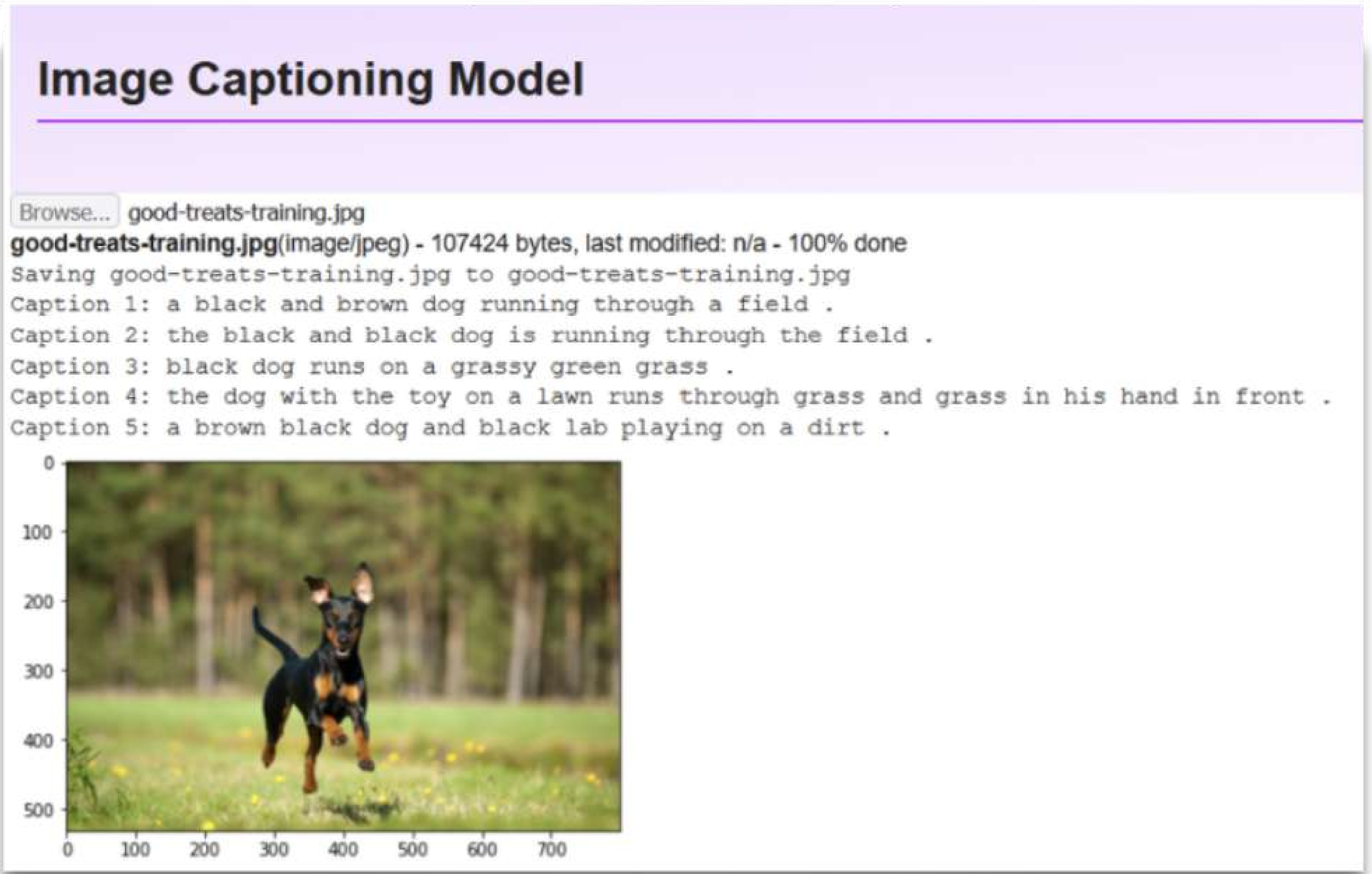


Image Captioning Model

Browse... good-treats-training.jpg

good-treats-training.jpg(image/jpeg) - 107424 bytes, last modified: n/a - 100% done

```
Saving good-treats-training.jpg to good-treats-training.jpg
Caption 1: a black and brown dog running through a field .
Caption 2: the black and black dog is running through the field .
Caption 3: black dog runs on a grassy green grass .
Caption 4: the dog with the toy on a lawn runs through grass and grass in his hand in front .
Caption 5: a brown black dog and black lab playing on a dirt .
```

Areas for further work......

- We could not run inceptionV3 with transformers and ResNet50+Transformers with image data augmentation due to GPU restrictions of colab.
- The vocabulary obtained from the flickr8K dataset was limited and restricted the predictive power of the model

Acknowledgements

We would like to extend our sincere gratitude to our PS Faculty Dr.Ramakrishna Dantu for his consistent support and motivation throughout the course of PS-1 and for his persistent efforts to get us onboarded to the company.

We are also thankful to our industry mentors Mr.Andrew Anand, Mr. Umesh and Mr. Vivek Ananthan for giving us this project which has enhanced our knowledge of machine learning and computer vision.

We would like to thank our project mentor Mr. Vivek Ananthan in particular for his exemplary guidance, monitoring and constant encouragement throughout the course of this project and for providing the necessary information and resources for this project. He has helped us a lot in improving our technical skills and got us accustomed to the corporate culture, helped us follow and stick to deadlines and give constructive suggestions to enhance our work.

Lastly we would like to thank our PS division and BITS Pilani for providing us this opportunity to develop our industrial knowledge and give us this enriching experience in the form of PS-1 that has not only helped us witness what we have been studying but also implement them in the real world.

.

References

[1]http://cs231n.stanford.edu/reports/2016/pdfs/364_Report.pdf
[2]https://www.ripublication.com/ijaer18/ijaerv13n9_102.pdf
[3]https://towardsdatascience.com/a-guide-to-image-captioning-e9fd5517f350
[4]Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin; *Attention is all you need*,2017
[5]https://datascience.stackexchange.com/questions/51065/what-is-the-positional-encoding-in-the-transformer-model
[6]https://towardsdatascience.com/illustrated-guide-to-transformers-step-by-step-explanation-f74876522bc0
[7]https://www.tensorflow.org/text/tutorials/transformer
[8]*ResNet-50 architecture [26] shown with the residual units, the size of the filters and the outputs of each convolutional layer.* (2020, January). [Graph]. Automatic Hierarchical Classification of Kelps Using Deep Residual Features.
[9]https://www.researchgate.net/figure/ResNet-50-architecture-26-shown-with-the-residual-units-the-size-of-the-filters-and_fig1_338603223
[10]https://www.pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/
[11]https://www.researchgate.net/figure/The-architecture-of-ResNet50-and-deep-learning-model-flowchart-a-b-Architecture-of_fig1_334767096

Thank you