The American University in Cairo

Image Caption Generator

Shahd Elmahallawy & Nermien Elassy

ABSTRACT

It is challenging for machines to interpret the description of an image. Captioning an image is a common computer vision task that is not only for capturing the objects in the image but also describes the relationships between them. Thus, it is a fundamental task in developing Artificial Intelligence.

Our motivation behind it is to help make the life of visually impaired people easier.

We choose a model by Al-Malla et al. It is very a recent model and showed good results. We used the same model but on another dataset which is Flicker 30k. We tired though this project to change some hyperparameters and edit in it aiming to reach better results.

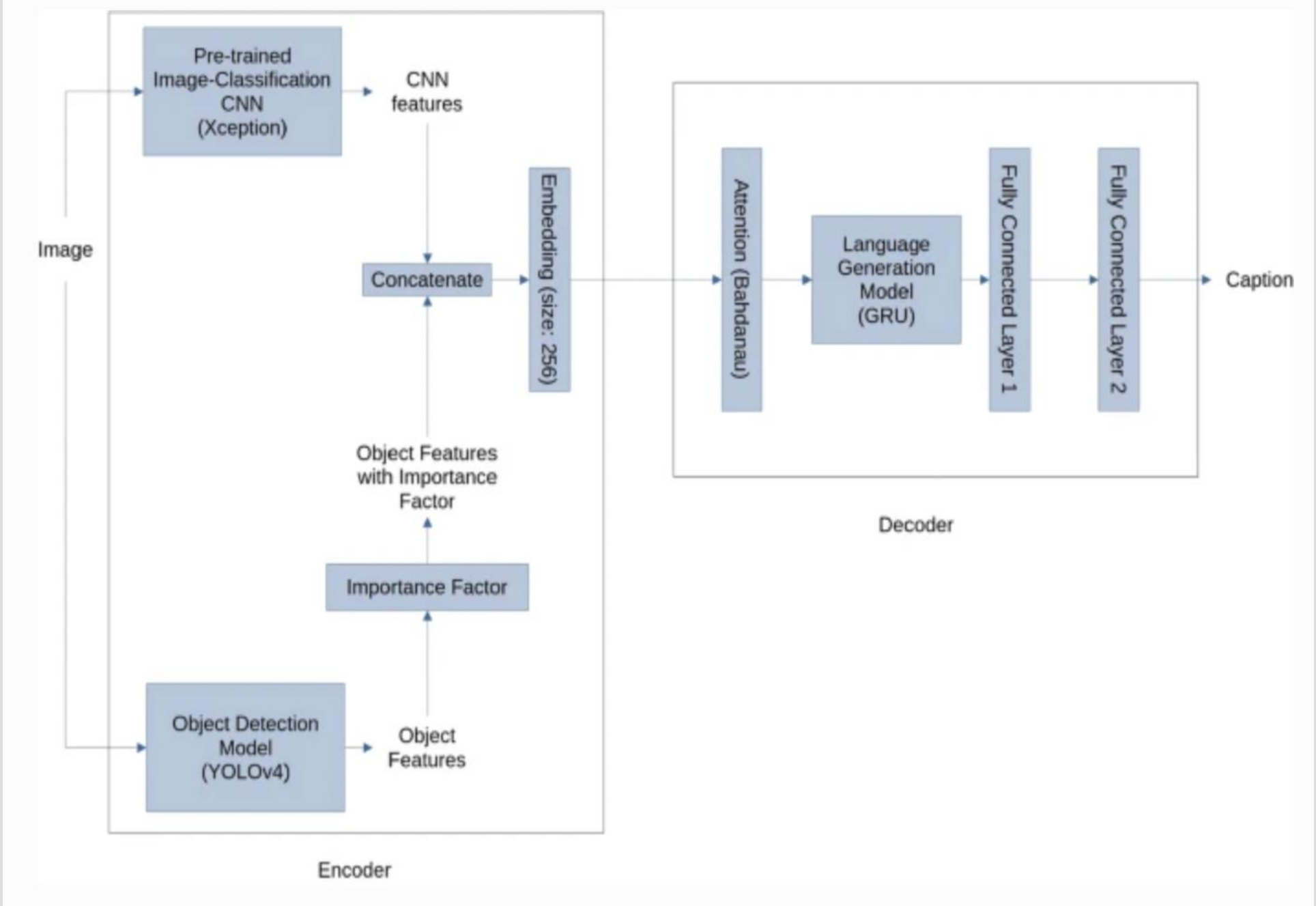
Problem Statement

The problem is to make machines more capable of using natural language processing technologies to understand images better with details.

Model

The model is an attention-based Encoder-Decoder architecture.

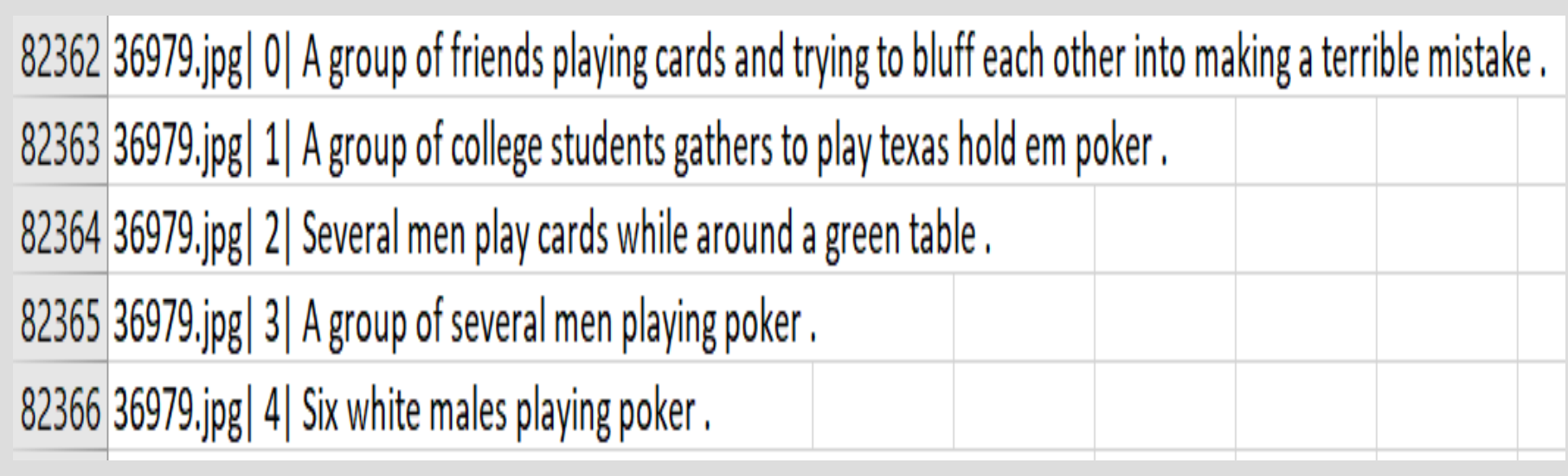
- The encoder part is a concatenation of two models:
  - YOLOv4 (object detection)
  - CNN (image classification).
- The decoder part uses two models
  - CNN (Xception)
  - Yolov4.
- Bahdanau (soft attention system) to highlight the input data's most significant parts and ignore the rest.
- Gated recurrent to produce caption is produced by one-word generation at every time step.
- Two fully connected layers.
  - The first layer has a length of 512.
  - The second layer has the size of the output text vocabulary.



Dataset

We used **Flicker30K** dataset.

- It is approximately 4.1 GB.
- It has about 31,783 images with 158,915 captions.
- Each Image has five captions as shown in the below example:

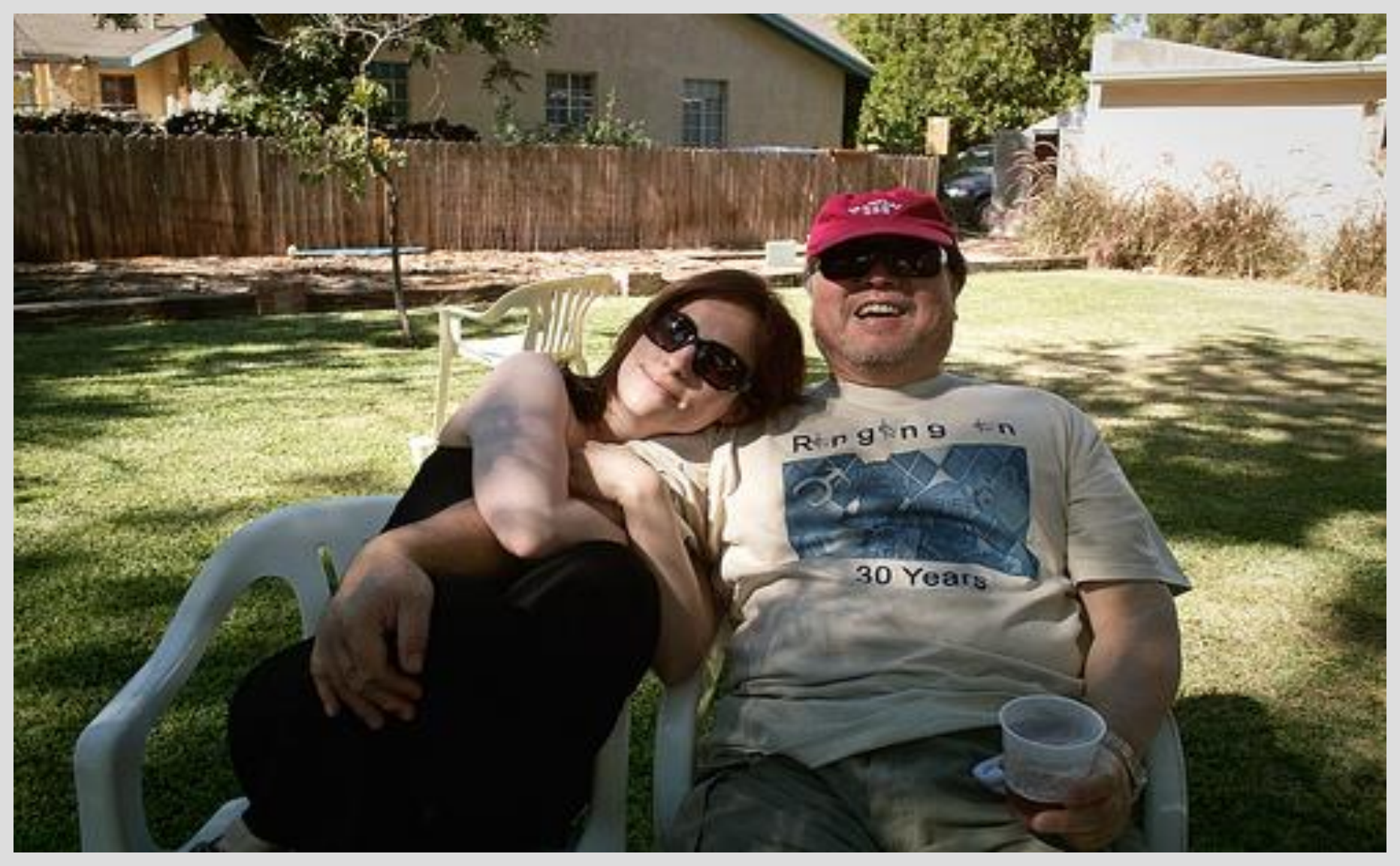


82362	36979.jpg| 0| A group of friends playing cards and trying to bluff each other into making a terrible mistake .
82363	36979.jpg| 1| A group of college students gathers to play texas hold em poker .
82364	36979.jpg| 2| Several men play cards while around a green table .
82365	36979.jpg| 3| A group of several men playing poker .
82366	36979.jpg| 4| Six white males playing poker .

RESULTS

Example of generated captions after running:

"image_id": 2975627633,
"caption": "a guy in an red hat and a child are sitting in a third girl"
"original_caption": "a woman is hugging a man 's arm"



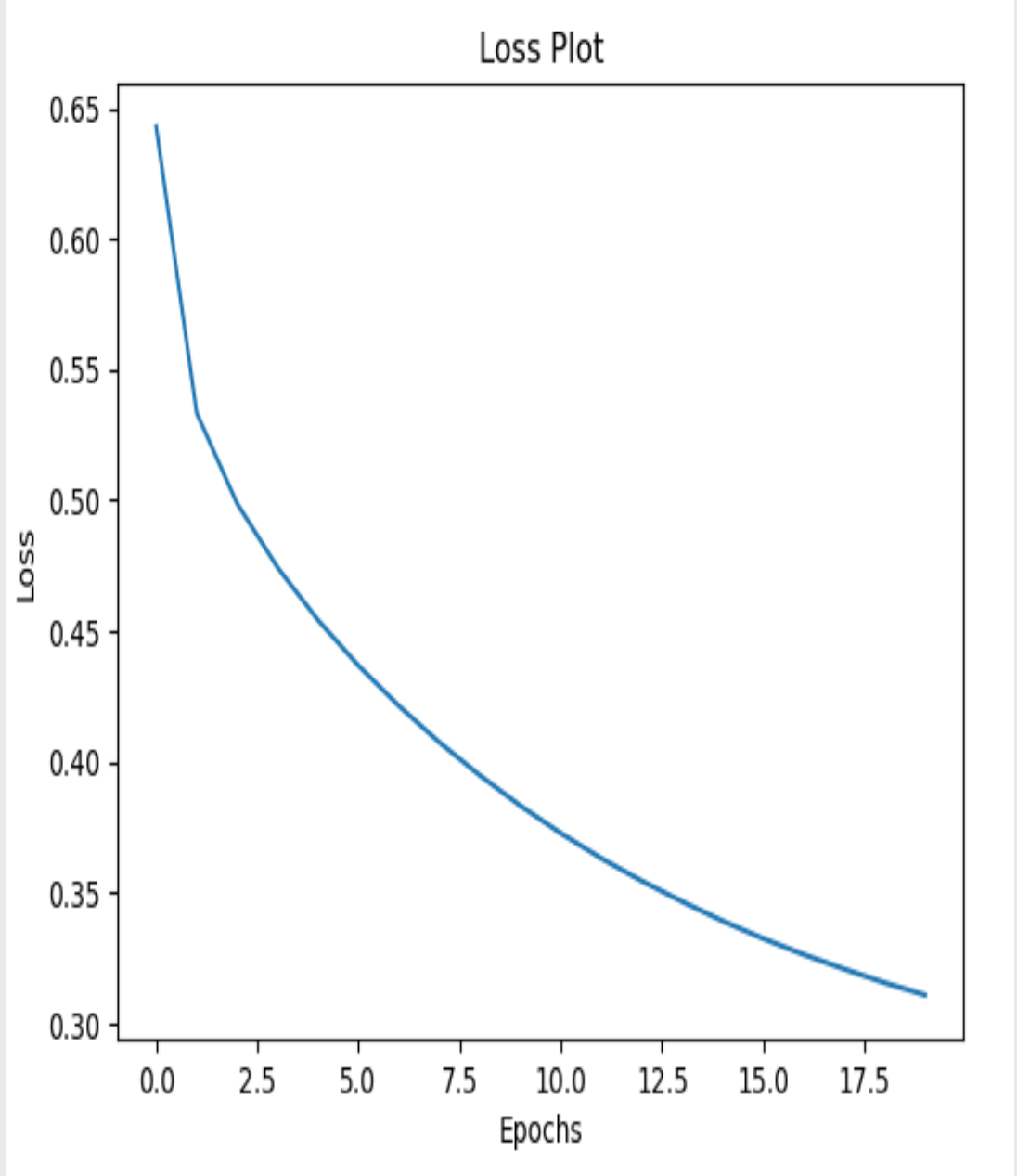


Figure 1. Deploying the same model but on Flicker 30k (Loss = 0.31)

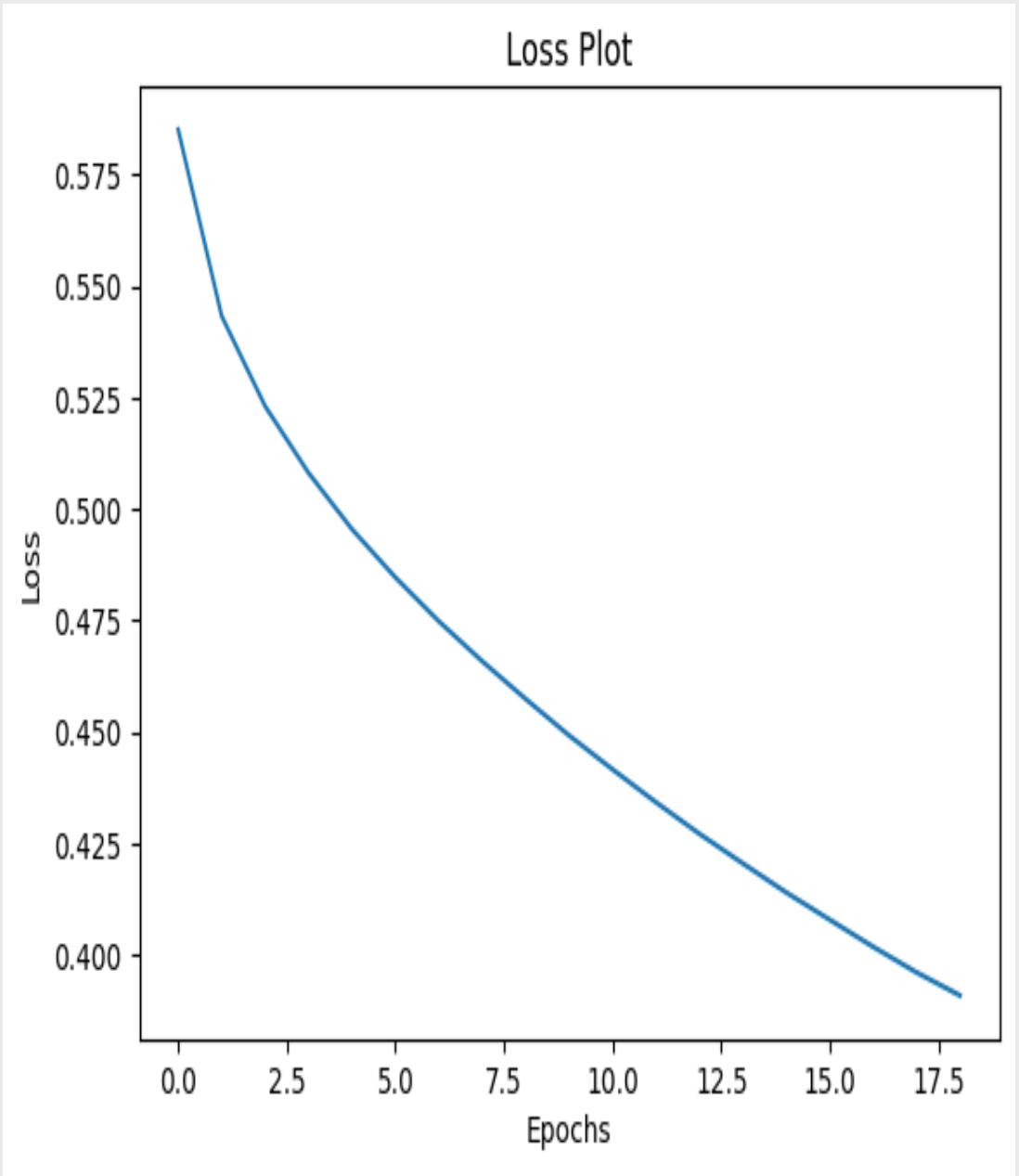


Figure 2. Changed CNN (Loss = 0.39).

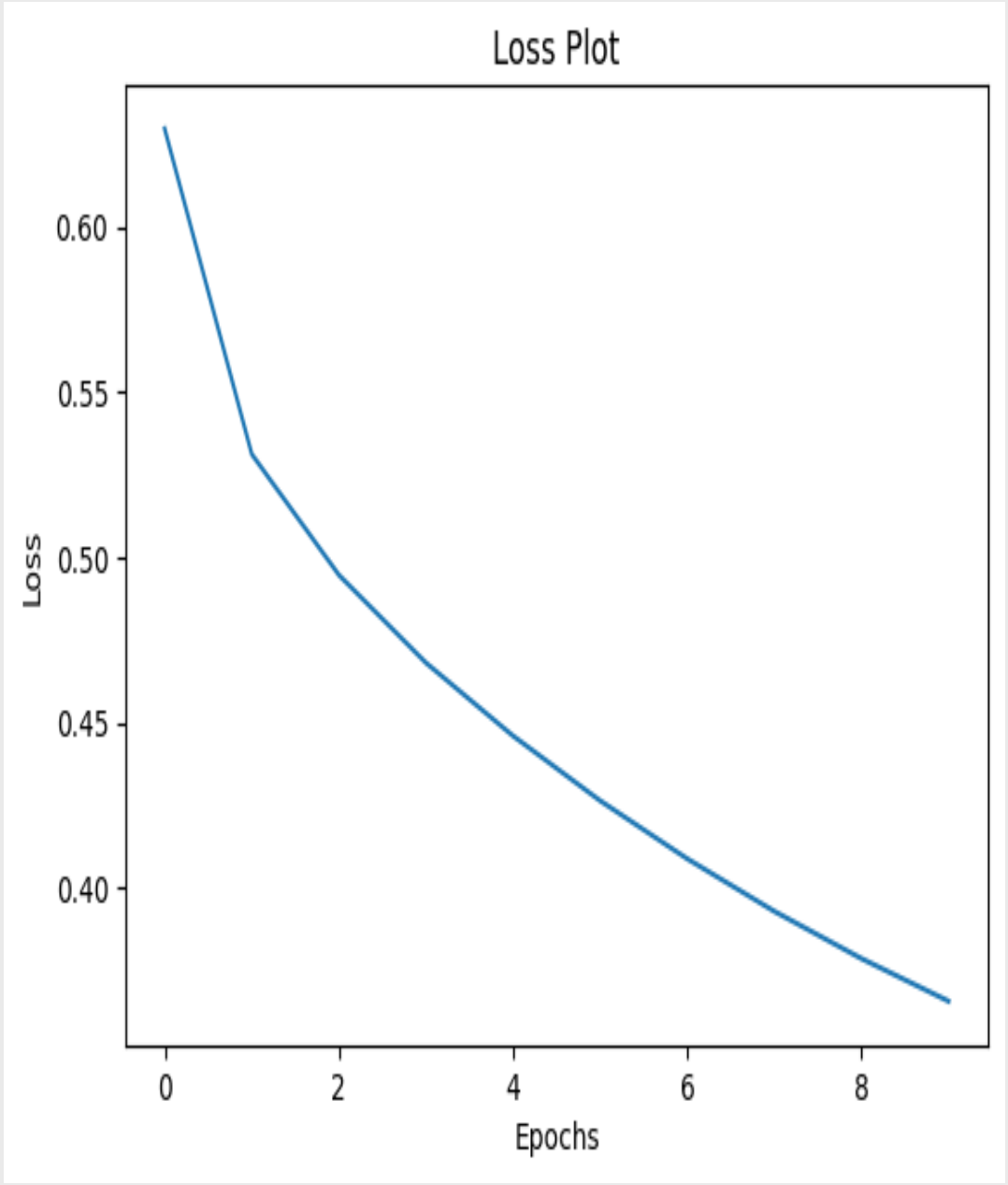


Figure 3. Added 5000 Augmented image. (Loss = 0.365)

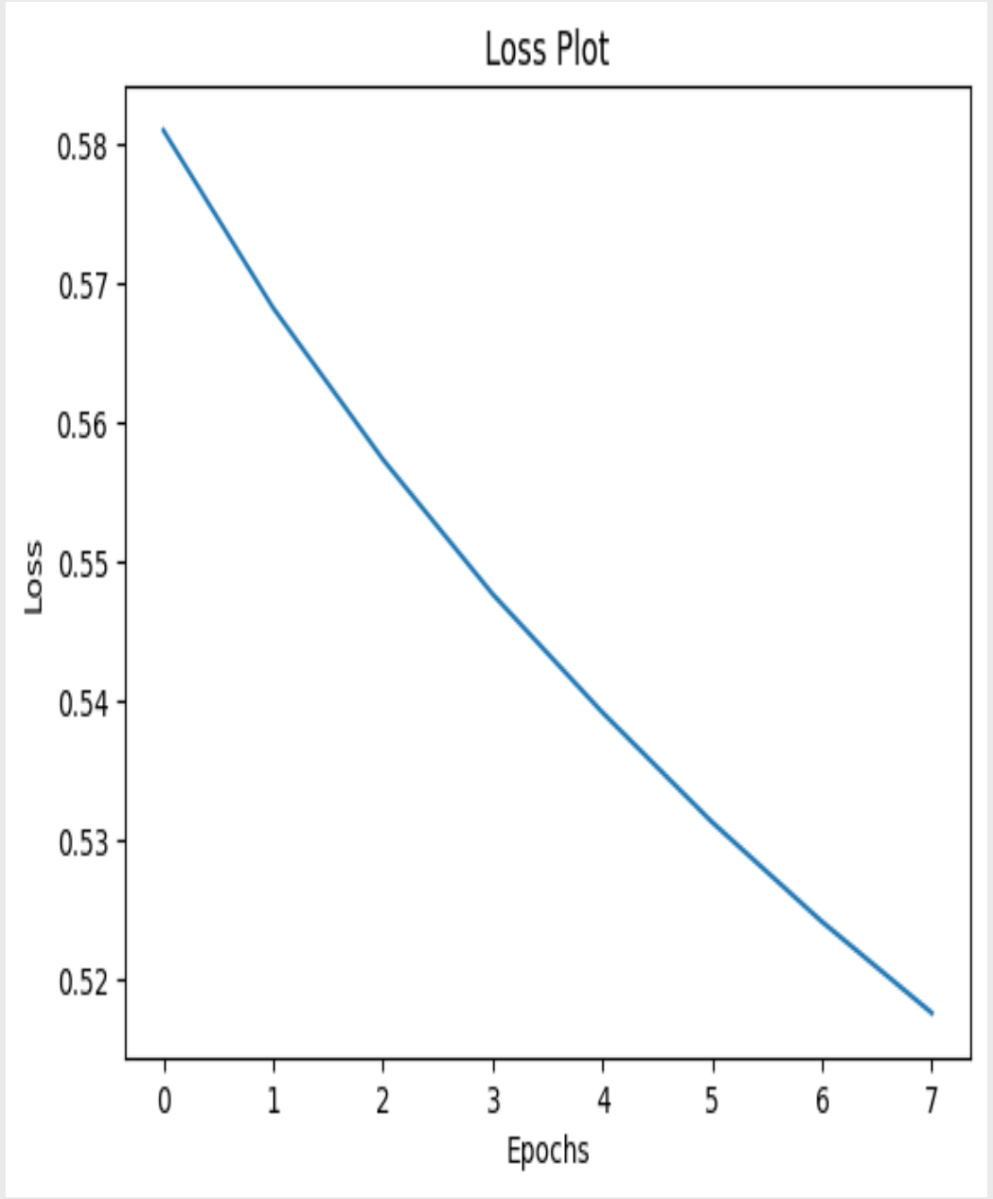
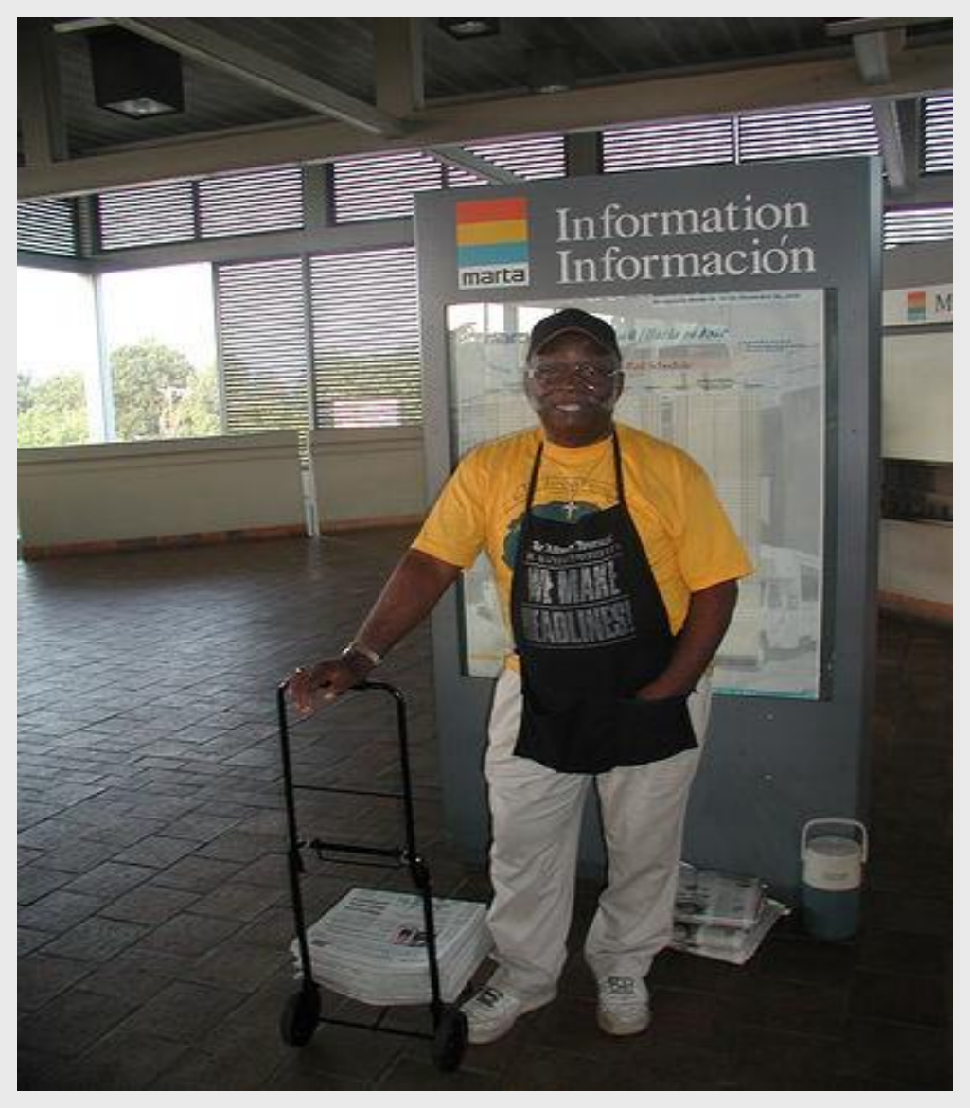
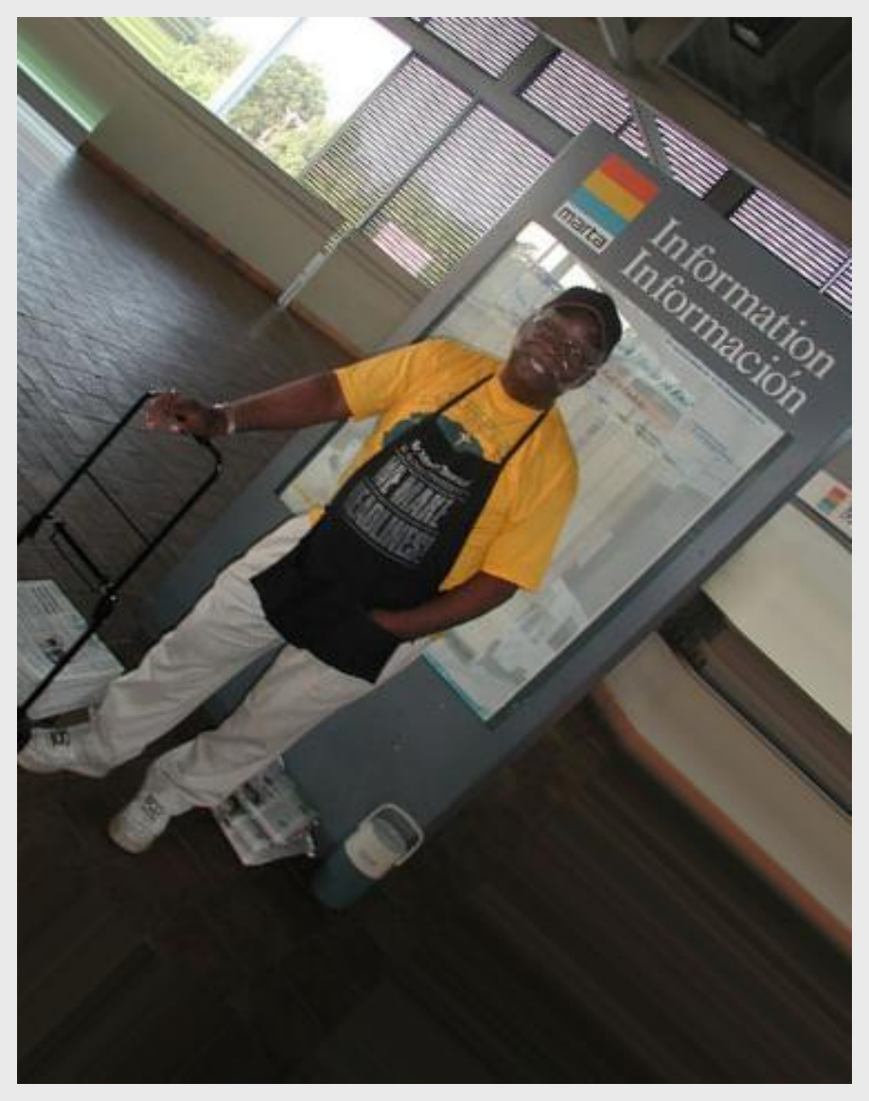


Figure 4. Change optimizer to SGD (Loss = 0.517)

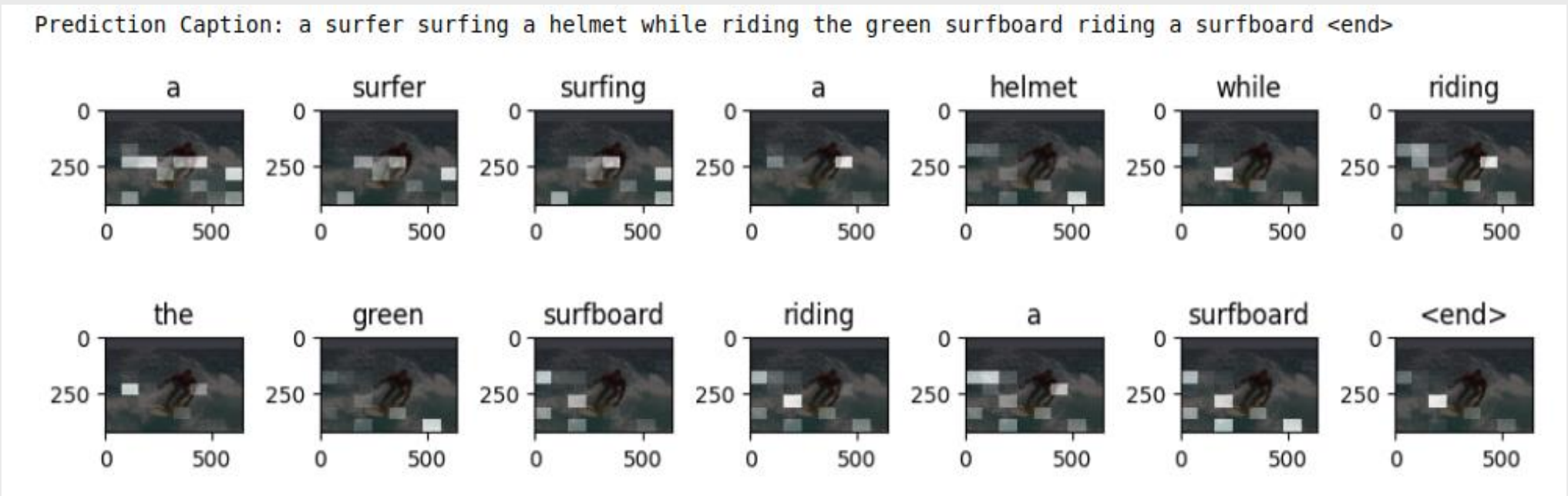
Example on Data Augmentation:
Image rotated by 90 Degrees



Original Image. 148284.jpg



Augmented. 148284.jpg_0_7113.jpg



Prediction Caption: a surfer surfing a helmet while riding the green surfboard riding a surfboard <end>

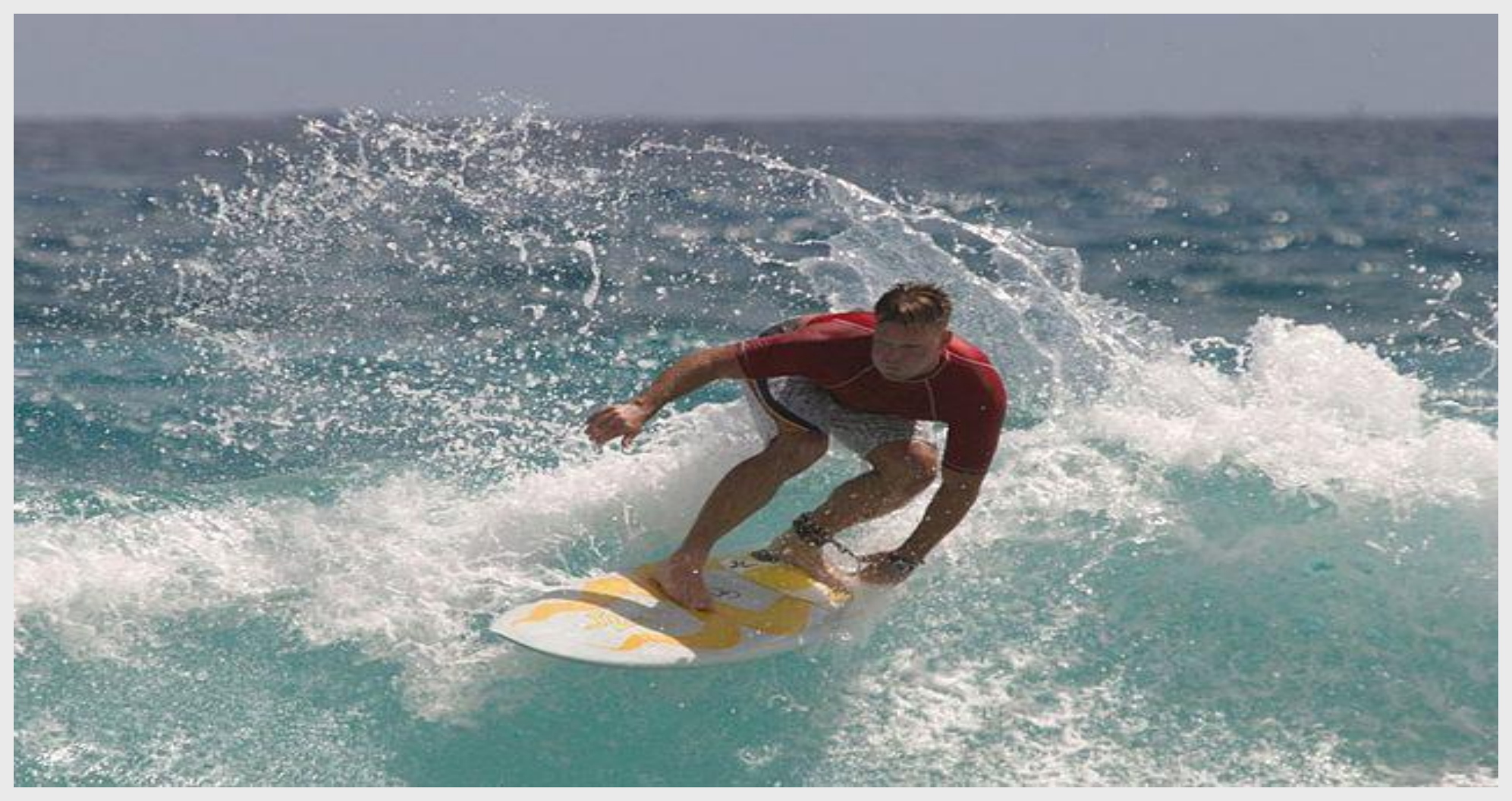


Figure 5. Test the model on an image not in the dataset. URL of the image: “https://tensorflow.org/images/surf.jpg”

DISCUSSION

We used first the original model but applied it on Flicker30K dataset. We then applied some changes in the hyperparameters. We found that after epoch 10 the results are almost the same. Then, we tied to add some augmented data and see the effect. We applied the data augmentation on only 5000 image by rotating he images, and it almost showed similar results. We change the CNN layer by adding one more fully connected layer and added activation functions (Relu, Softmax). We tuned some hypermeters such as the number of epochs, different optimizers, and learning rate. We noticed that after Epoch 10, the results were almost the same, so we changed to 10.

CONCLUSION

All the runs showed small differences in the accuracy. The best accuracy was achieved Changing gin the CNN model showed
So, we considered as our final model; however, we didn’t manage to achieve accuracy better than the original model.

For future work, we think we can customize the evaluation metric for this model using flicker 30k since there is a huge drawback in this dataset that prevents it from being evaluated using the code that was designed for coco dataset. Also, the accuracy could be increased doing more hyperparameter tuning and using more data.

REFERENCES

[1] https://doi.org/10.1186/s40537-022-00571-w.

[2]https://github.com/abdelhadie-almalla/image_captioning

[3]http://hockenmaier.cs.illinois.edu/DenotationGraph/

[4]https://github.com/kiyoshiiriemon/yolov4_darknet