Fast R-CNN and Multimodal Attention Architecture for Image Captioning

Armeen Ahmed
Stanford University
armeen@stanford.edu

Kyle Schmoyer
Stanford University
kyles7@stanford.edu

Abstract

This paper proposes a Fast Region-based Convolutional Network method (Fast R-CNN) for object classification from images. We aim to develop an image captioner to aid visually impaired children by providing narratives of their surroundings through images. The proposed model leverages Fast R-CNN for its efficiency in detecting and classifying objects within each image. To enhance the model's ability to focus on relevant elements within these dynamic scenes, we incorporate a multimodal attention mechanism. This mechanism intelligently allocates computational resources towards significant visual cues, effectively mimicking the natural human attention process. This, in conjunction with textual transformers will allow for narrative sounding captions that can tell a story about the world around them.

1. Introduction

Over 20 million people in the US alone suffer from vision issues that impair them from seeing the natural word around them [7]. There are over 1.5 million blind children worldwide, and a child is born blind approximately every minute. With the advent of AI, we can now automatically classify and caption images, giving more context to visually impaired people's everyday lives. Previously, this would take hundred of hours for people to caption images for the purpose of accessibility. In this, we will combine image captioning with the latest large language models in order to better convey image context to those with visual impairments. This will allow for a more narrative sounding description, rather than the short captioning style that models typically produce. In this we both want our captioner to be aware of specific objects within the image and the background situational awareness. For this, we will use a faster R-CNN to encode images. We then plan to use a multimodal self attention mechanism to decode following the faster R-CNN. We compare a pre-existing model, YOLO, for a baseline performance on the dataset. For our multimodal self attention mechanism, we will be utilizing Meta AI's FLAVA to preform semantic segmentation. The input to our algorithm will be an image, and we will then use a Faster-RCNN to output text describing the image.

2. Problem Statement

The COCO (Common Objects in Context) dataset classes are divided into two main categories: 'things' and 'stuff.' 'Things' classes include objects easily picked up or handled, such as animals, vehicles, and household items. Examples of 'things' class objects include 'person', 'bicycle', 'car', etc. 'Stuff' classes include background or environmental items such as 'sky', 'water', and 'road'. Although the COCO dataset can be adapted to various computer vision tasks, we chose to go for semantic segmentation, which entails detecting and segmenting objects and backgrounds within an image, encompassing both "things" (specific objects) and "stuff" (indistinct areas of the image like sky, water, and road). This allows for models to integrate both the material properties of the background, to allow for a greater description of background and context for the captions. In the context of the COCO dataset, semantic segmentation annotations provide complete scene segmentation, identifying items in images based on 80 "things" and 91 "stuff" categories which will allow blind users to fully comprehend their surroundings. These surroundings are very important to convey the location of a particular image, which will assist blind users to better understand the background.

3. Literature Review

We explored existing literature and research journals that also attempted image captioning using various methods.

3.1. Transformer based Multitask Learning for Image Captioning and Object Detection

This research done by Basak et al. explores the process of captioning images utilizing a faster R-CNN in conjunction with a Swin transformer background[3]. This uses a combination of loss functions that improved their preformance. Finally, they combined the output with GPT-2 in order form comprehensive captions for images.

3.2. Multi-Modal Image Captioning for the Visually Impaired

This research done by Ahsan et al. explore the use of Multi-Modal image captioning on a large dataset [11]. They also utilize Optical Character Recognition in their model to tokenize words that may be in the image to give an image more context. For this they modified the existing AOA-net model and saw large performance improvements.

3.3. Visuals to Text: A Comprehensive Review on Automatic Image Captioning

This research done by Ming et al. explores the use of different encoder and decoder attention methods [4]. It also provides a comprehensive overview of existing methods, and their performance on commonly found datasets.

3.4. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering

This research done by Anderson et al. combines both bottom-up and top-down attention mechanisms to improve performance. The bottom-up portion proposes the object regions while the top-down portion improves attention based on performance. This found improvements in performance by focusing computation on the important portion of images [1]. This was quite clever and state of the art at the time.

3.5. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

This research done by Xu et al. proposed an early attention-based model that is able to refocus its attention as each word is generated. This allowed for much more descriptive captions at the time and sparked a focus on attention-based models [14].

3.6. UNITER: UNiversal Image-TExt Representation Learning

This research done by Chen et al. proposed a unified transformer model for joint image-text learning. This allowed for it to be capable of handling multiple tasks, such as captioning and visual question answering. This model was also pre-trained on large datasets, and allowed for ease of use [2].

3.7. Show and Tell: A Neural Image Caption Generator

This research done by Vinyals et al. was one of the first to propose an encoder-decoder framework to caption images. They utilized a CNN encoder for images and then an RNN to decode into descriptive captions [13].

3.8. A Comprehensive Survey of Deep Learning for Image Captioning

This research done by Hossain et al. provides an overview of different techniques for image captioning, covering encoder-decoder architectures, attention mechanisms, and evaluation metrics. It also provides strengths and weaknesses of existing methods [6]. This is a strong paper that serves as the basis for many others that look to improve upon the methods described within.

3.9. Pointing Novel Objects in Image Captioning

This research done by Li et al. highlights the challenge of encountering objects that were not seen during training. They propose using an attention mechanism to focus on image regions that would allow the model to generate contextual captions for novel objects [9].

3.10. Unified Vision-Language Pre-Training for Image Captioning and VQA

This research done by Zhou et al. proposes a unified pre-training approach. By using this single model, they saw large improvements in performance, and highlighted the importance of multi modal learning [15]. This was a quite strong approach, and a model example for the performance of FLAVA.

4. Dataset

For the purpose of the task, we considered several popular computer vision datasets and decided to use COCO (Common Objects in Context), which is a large-scale object detection, segmentation, and captioning dataset. It contains over 330,000 images, each annotated with 80 object categories and 5 captions describing the scene [10]. Other popular datasets for the image classification problem include CIFAR-10 and CIFAR-100, providing 10 and 100 categories respectively, from a collection of small 32×32 pixel images. However, despite encompassing as many as 60,000 images and spanning hundreds of categories, these datasets represent only a minor segment of the vast diversity of our visual environment. Another popular dataset we considered was the ImageNet dataset. Upon scrutinizing the ”benchmark task misalignment” in ImageNet, the team from MIT discovered that approximately 20% of the images in ImageNet contain several objects [12]. Their analysis, spanning various object recognition models, indicated that the presence of multiple objects in a single photo could cause a general accuracy decline of about 10%. Compared to ImageNet, COCO features fewer categories but includes a greater number of instances within each category. This characteristic can facilitate the development of detailed object models that excel in precise 2D localization.

Figure 1. Examples from the COCO dataset with labelling.

For training our model, we used the 2017 dataset split. We used the custom dataset class COCODataset, which interfaces with the COCO API to load images and their annotations dynamically. The dataset ensures each image is loaded with its respective bounding box and category labels for segmentation.For preprocessing, we converted PIL images into PyTorch tensors, which are suitable for input into PyTorch models and normalized the pixel values for stability and faster convergence during training.

To illustrate the data and the effectiveness of our preprocessing and feature extraction, below are examples of an original image from the COCO dataset alongside its corresponding segmentation mask generated by our model. The segmentation mask highlights the model's capability to distinguish between different object categories and outlines within the same image.

Figure 2. Original COCO Image

Figure 3. Corresponding Segmentation Mask

The feature extraction process was directly integrated with the model architecture. The backbone of our model, a modified ResNet50 network, was responsible for extracting high-dimensional features from the input images. These features were then passed to the Region Proposal Network (RPN) and ROI align layers for generating and refining object proposals.

We did not use traditional feature extraction techniques such as Fourier transforms, HOG, or PCA. Instead, the deep learning model learned to extract and refine features automatically during training, which is more effective for complex tasks like semantic segmentation.

5. Method

To achieve our goals, we test a variety of different encoders that utilize the faster R-CNN model. For this we will utilize a pre-trained model, Res-Net50 and further fine tune the model on our dataset. The choice of ResNet50 as the backbone for the Faster R-CNN model in the context of object detection and semantic segmentation is crucial due to its robust feature extraction capabilities. ResNet50 is a variant of the Residual Network architecture that includes 50 layers deep, renowned for its ability to handle very deep neural networks without succumbing to the vanishing gradient problem. This is achieved through the use of residual connections that add outputs from previous layers to the outputs of stacked layers, thus enabling training of much deeper networks by facilitating the flow of gradients.

In the typical deployment within a Faster R-CNN framework, the last fully connected layers of ResNet50 are removed, and instead, the feature maps generated by the earlier convolutional layers are used. Removing the final global pooling and fully connected layers helps in retaining the spatial resolution of the feature maps, which is crucial for accurately localizing objects within an image. The higher resolution feature maps contain more detailed spatial information that is beneficial for generating precise region proposals in subsequent stages of the Faster R-CNN. o adapt the output of the ResNet50 to fit into the next stages of the Faster R-CNN, the channels of the output feature maps may be modified. This ensures compatibility with the Region Proposal Network (RPN) and ROI align layers.

After feature extraction through the modified ResNet50 backbone, the feature maps are fed into the RPN. The primary function of the RPN is to generate object proposals within the feature map, which are candidate regions where objects might be located. The RPN utilizes an anchor generator to create multiple anchor boxes at each location of the feature map. These anchors serve as reference boxes to which the ground truth objects are compared during training. Each combination of size and aspect ratio at each spatial location on the feature map yields a dense coverage of anchor boxes, ensuring that all parts of the image are

scanned for potential objects.

Once the RPN proposes regions likely to contain objects, the next step is to extract a fixed-size small feature map from each region proposal for further processing. This is accomplished using the ROI Align technique. The ROI Align technique takes these proposals and extracts fixed-size feature sections from the feature maps for each proposal. These features are subsequently used by the Faster R-CNN's classifier and bounding box regressor to predict the class and adjust the coordinates of each object, respectively. In the final stages of the Faster R-CNN, the features extracted via ROI Align are passed to the head network. The head network consists of two main components - classifier and bounding box regressor. This network head uses the ROI-aligned features to determine the class of each proposed object region. It outputs class probabilities for each region, indicating the likelihood of each class being present. Alongside classification, this head adjusts the coordinates of the initially proposed bounding box to better fit the actual object. It outputs refinements for the location and size of each box, thereby improving the precision of the object localization.

We used the Adam optimizer, known for its effectiveness in handling sparse gradients and non-stationary objectives, which are common in deep learning tasks like object detection. The initial learning rate is set to 0.0001. This rate is crucial as it determines the step size at each iteration while moving toward a minimum of the loss function. A learning rate scheduler with a step size of 3 is used so very 3 epochs, the learning rate is decreased by $\gamma = 0.1$. The weight decay is set at 0.0005 to regularize the model by penalizing large weights, which can prevent overfitting to the training data.

The DataLoader is set up with a batch size of 1, meaning each batch consists of a single image and its corresponding annotations. While larger batch sizes can provide smoother gradient estimates, a batch size of 1 (often used in object detection due to memory constraints) ensures that the model can handle high-resolution images and complex annotations.

We also integrated a Vision Encoder-Decoder architecture that combines the Vision Transformer (ViT) with GPT-2, a transformer-based model for natural language processing using the Hugging Face transformers library. This hybrid model, pre-trained on a diverse dataset, is designed to handle tasks that require an understanding of both visual content and language generation. ViT applies self-attention mechanisms to the entire image, processing it as a sequence of patches. These patches are then linearly embedded, combined with positional encodings, and fed into the transformer encoder. The encoder outputs a set of feature vectors that represent various aspects of the visual input. This allows it to capture contextual relationships between different parts of the image. GPT-2 takes the encoded features from ViT and decodes them into descriptive text, maintaining logical and grammatical coherence. GPT-2 generates captions by predicting one word at a time. This is done through a sequence-to-sequence model where each subsequent word depends on the previously generated words.

Additionally, we explored multimodal attention based architectures, including the FLAVA model [5]. This model allows for the conjuction of both text and video encoders, which would improve preformance greatly. Additionally, speed is one of our top priorities, in order quickly generate captions for those who are visually impaired. Our final step of optimization is the loss. We test a variety of losses, such as cross entropy loss as shown below.

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(p_{i,c})$$

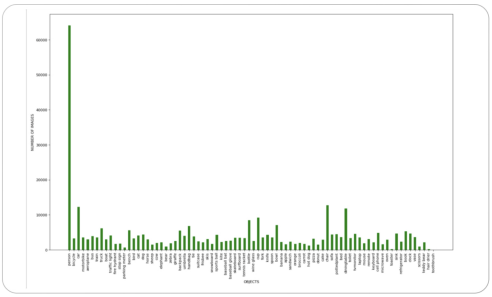

Figure 4. Class imbalance in the COCO dataset

Additionally, the focal loss method could be useful because of its high performance on datasets with class imbalance. Class imbalance occurs when there is a significant disparity in the number of samples across different classes. Within the context of the COCO dataset, certain object classes are represented by a much larger number of image instances compared to others. This is illustrated in the chart above.

$$\mathcal{L}_{\text{FL}} = -\alpha_t(1-p_t)^{\gamma}\log(p_t)$$

Now that we have described our proposed model, let us take a look at our baseline model we will be using for comparison. Our baseline model was YOLOv5 developed by Ultralytics, a cutting-edge SOTA model in the You Only Look Once (YOLO) family of computer vision models. It is an extremely fast object detection framework using a single convolutional network.

The backbone of YOLOv5 is based on the CSPDarknet53, which is a variation of the Darknet architecture used in earlier YOLO versions. CSPDarknet introduces Cross Stage Partial connections (CSP), which help in reducing the computational cost and improving the learning capability of the model. This backbone processes the input im-

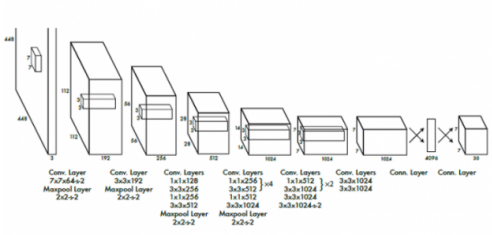

Figure 5. Example of Yolo architecture for baseline.

age to extract feature maps that capture various levels of detail, which are then used by subsequent layers for detecting objects. Following the backbone, YOLOv5 uses a neck based on the Path Aggregation Network (PANet) architecture. This part enhances the feature hierarchy via a bottom-up path augmentation, which facilitates the propagation of lower-level features to higher-level layers. The head of YOLOv5 is responsible for making the final object detection predictions. It uses anchor boxes to predict bounding boxes relative to the anchors. For each anchor, the model predicts four coordinates for the bounding box, one objectness score, and several class probabilities (depending on the number of classes in the task). The objectness score predicts the likelihood of an object being present in the bounding box, while the class probabilities determine what object is in the bounding box.

Finally for the FLAVA model, a multimodal attention architecture developed by meta combines both visual and textual encoders. FLAVA uses these vision transformers to segment and process images as patches, capturing important details. This is packaged with a transformer-based language model that process this data and creates descriptions. We fine-tuned a pre-trained FLAVA model with the coco dataset. This improved the accuracy in recognizing the 80 objects in the dataset, and focal loss allows for it to handle difficult examples.

Device Utilization (CPU/GPU) The model utilizes CUDA if available, which implies that if a compatible GPU is present, it will be used for computation. The use of a GPU significantly accelerates the training and inference processes due to parallel processing capabilities, which are particularly beneficial for the computationally intensive operations involved in deep learning models like Faster R-CNN.

6. Metrics and Results

We evaluated the performance of our models using three metrics - precision, recall and F1 score. Precision refers to the number of true positives divided by the total number of positive predictions (i.e., the number of true positives plus the number of false positives).

$$\text{Precision} = \frac{TP}{TP + FP}$$

where,

- TP (True Positives) is the number of correct positive predictions made by the model.
- FP (False Positives) is the number of negative instances incorrectly predicted as positive.

Recall, also known as the true positive rate (TPR), is the percentage of data samples that a machine learning model correctly identifies as belonging to a class of interest—the "positive class"—out of the total samples for that class.

$$\text{Recall} = \frac{TP}{TP + FN}$$

where,

- TP (True Positives) is the number of correct positive predictions made by the model.
- FN (False Negatives) is the number of positive instances incorrectly predicted as negative.

The F1 score is a measure of the harmonic mean of precision and recall. Maximizing for the F1 score implies simultaneously maximizing for both precision and recall.

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision + Recall}}$$

We calculated these metrics for individuals categories as well as their averages in our baseline model (YOLOv5) and our proposed model (Faster R-CNN), which are illustrated in the tables below. To further enhance our understanding of the model's performance across different categories, we complemented our evaluation with confusion matrices. These matrices provide a visual representation of the accuracy of the model by displaying the actual versus predicted classifications for each category.

For the FLAVA model, we utilized the BLEU-4 scoring metric in order to determine its preformance, which resulted in a score of 32.4. Reviewing the outputs of both trained models, the FLAVA model seemed much more coherent in its captions, and performed better than the Faster R-CNN for captioning.

	Precision	Recall	F1-Score
person	0.93	0.68	0.79
car	0.86	0.59	0.70
chair	0.77	0.39	0.52
book	0.71	0.08	0.14
bottle	0.82	0.49	0.62
cup	0.82	0.56	0.67
dining table	0.75	0.23	0.35
traffic light	0.81	0.47	0.59
bowl	0.73	0.49	0.58
handbag	0.71	0.23	0.35
micro avg	0.88	0.56	0.68
macro avg	0.79	0.42	0.53
weighted avg	0.87	0.56	0.67

Table 1: Yolov5 Performance Metrics

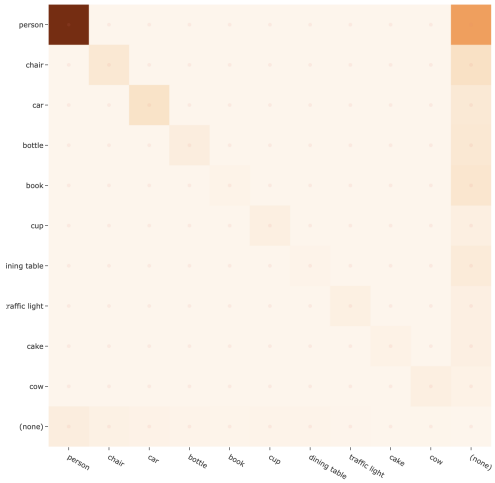

Figure 6. Yolo Confusion Matrix for 10 Classes

	Precision	Recall	F1-Score
person	0.77	0.84	0.80
car	0.62	0.72	0.67
chair	0.51	0.48	0.50
book	0.53	0.61	0.57
bottle	0.53	0.63	0.57
cup	0.54	0.61	0.57
dining table	0.38	0.53	0.44
traffic light	0.54	0.61	0.57
bowl	0.50	0.67	0.58
handbag	0.34	0.29	0.31
micro avg	0.66	0.73	0.70
macro avg	0.53	0.60	0.56
weighted avg	0.66	0.73	0.70

Table 2: Faster R-CNN Performance Metrics

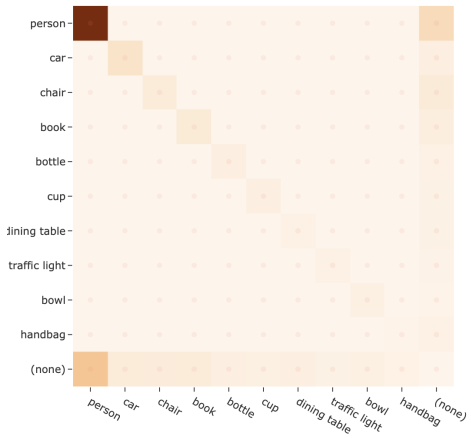

Figure 7. Faster R-CNN Confusion Matrix for 10 Classes

7. Discussion

The overall performance, as indicated by the micro and macro averages, shows that YOLOv5 exhibits higher precision (0.88 micro average) compared to Faster R-CNN (0.66 micro average), suggesting that YOLOv5 is more accurate in its predictions when it positively identifies objects. However, Faster R-CNN demonstrates superior recall (0.73 micro average) over YOLOv5 (0.56 micro average), indicating that Faster R-CNN is more effective at identifying relevant objects within the images. The weighted F1-score, which balances precision and recall, is marginally higher for Faster R-CNN (0.70) compared to YOLOv5 (0.67), suggesting that when considering both precision and recall, our proposed model, Faster R-CNN may provide a more balanced performance overall.

If we do a category-specific analysis, for objects like 'person', Faster R-CNN outperforms YOLOv5 in terms of recall and F1-score, which is crucial for applications where missing a 'person' in the image could lead to significant repercussions, such as in surveillance systems. Conversely, the 'handbag' and 'dining table' categories show relatively poor performance across both models, but particularly in Faster R-CNN, where precision and recall are significantly lower than YOLOv5. This suggests difficulty in detecting smaller or less distinct objects, which could be attributed to variations in object size and occlusions within the training data.

The trade-off between precision and recall could have important implications on model selection. While YOLOv5 provides high precision, its lower recall might limit its utility in applications where missing objects is critical. Faster R-CNN, although less precise, offers better coverage in detecting objects. For instance, applications that require high precision and speed, such as real-time object tracking, might favor YOLOv5. Conversely, applications that cannot afford to miss objects, like automated monitoring systems

or automous vehicles, might benefit from Faster R-CNN's superior recall.

8. Conclusion and Future Work

Our proposed model, the Faster R-CNN had a higher weighted F1 score and a superior balance between recall and precision when compared to our baseline model, YOLOv5. This suggests that Faster R-CNN could serve as a more holistic model for applications requiring robust object detection capabilities.

However, the task of perceiving the natural world around us is not an easy one. While our proposed model performs well, we recognize its limitations. Future work on this topic would be panoptic segmentation models which unify the typically disparate tasks of semantic segmentation (assign a class label to each pixel) and instance segmentation (detect and segment each object instance). Panoptic segmentation requires generating a coherent scene segmentation that is rich and complete, closely resembling how humans perceive their environments. This approach is especially promising for developing advanced vision systems to assist the blind. Previous work on PS is based on the heuristic combination of outputs from top- performing instance and semantic segmentation systems [8]. However, there is a clear need for groundbreaking research in developing end-to-end models that simultaneously address both semantic and instance segmentation. Advancements in this area could challenge current methodologies and open new avenues for research and application, enhancing visual perception systems for assistive technologies and broader automation applications. This future direction not only aims to improve the technological landscape but also promises significant contributions to how machines interact and interpret complex visual environments. Future work towards this topic is important for impacting the lives of those with visual impairments and their ability to interpret the world around them.

References

[1] F. B. J. M. . G. S. Anderson, P. Bottom-up and top-down attention for image captioning and visual question answering. 2018. 2

[2] C. Y. Y. Y. C. Y. W. F. . Z. M. Chen, J. Uniter: Universal image-text representation learning. 2020. 2

[3] B. et al. Transformer based multitask learning for image captioning and object detection. *arXiv preprint arXiv:2403.06292*, 2024. 1

[4] M. et al. Visuals to text: A comprehensive review on automatic image captioning. *IEEE/CAA Journal of Automatica Sinica*, 9(5):105734, 2022. 2

[5] S. et al. Flava: A foundational language and vision alignment model. *arXiv preprint arXiv:2112.04482*, 2021. 4

[6] S. F. S. M. F. . L. H. Hossain, M. Z. A comprehensive survey of deep learning for image captioning. 2019. 2

[7] HPI Georgetown. Almost 20 million americans — 8 1

[8] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 7

[9] Y. T. P. Y. C. H. . M. T. Li, Y. Pointing novel objects in image captioning. 2019. 2

[10] M. M. B. S. H. J. P. P. R. D. D. P. . Z. C. L. Lin, T.-Y. Microsoft coco: Common objects in context. 2014. 2

[11] ReadKong. Multi-modal image captioning for the visually impaired, 2024. 2

[12] VentureBeat. Mit researchers find 'systematic' shortcomings in imagenet data set. 2024. 2

[13] T. A. B. S. . E. D. Vinyals, O. Show and tell: A neural image caption generator. 2015. 2

[14] B. J. K. R. C. K. C. A. S. R. Z. R. . B. Y. Xu, K. Show, attend and tell: Neural image caption generation with visual attention. 2015. 2

[15] P. H. Z. L. H. H. C. J. . G. J. Zhou, L. Unified vision-language pre-training for image captioning and vqa. 2020. 2

9. Contributions

A.A. researched and implemented baseline models, trained Faster R-CNN, wrote introduction, methodology, and conclusion, and generated visualizations K.S. trained Faster R-CNN and FLAVA model, wrote Literature review.