Enhanced Image Captioning with Color Recognition Using Deep Learning Methods

Yeong-Hwa Chang 1,2,* , Yen-Jen Chen 1, Ren-Hung Huang 1 and Yi-Ting Yu 1

1 Department of Electrical Engineering, Chang Gung University, Taoyuan City 333, Taiwan; M0721022@cgu.edu.tw (Y.-J.C.); M0721003@cgu.edu.tw (R.-H.H.); M0721021@cgu.edu.tw (Y.-T.Y.)
2 Department of Electrical Engineering, Ming Chi University of Technology, New Taipei City 243, Taiwan
* Correspondence: yhchang@mail.cgu.edu.tw

Abstract: Automatically describing the content of an image is an interesting and challenging task in artificial intelligence. In this paper, an enhanced image captioning model—including object detection, color analysis, and image captioning—is proposed to automatically generate the textual descriptions of images. In an encoder–decoder model for image captioning, VGG16 is used as an encoder and an LSTM (long short-term memory) network with attention is used as a decoder. In addition, Mask R-CNN with OpenCV is used for object detection and color analysis. The integration of the image caption and color recognition is then performed to provide better descriptive details of images. Moreover, the generated textual sentence is converted into speech. The validation results illustrate that the proposed method can provide more accurate description of images.

Keywords: image caption; color recognition; LSTM; object detection

1. Introduction

Image captioning essentially comprises two tasks: computer vision, and natural language processing (NLP). Computer vision helps to recognize and understand the scenario presented in an image, and NLP converts this semantic knowledge into a descriptive sentence. Automatically retrieving the semantic content of an image and expressing it in a form that humans can understand is quite challenging. The overall image captioning model not only provides the information, but also shows the relationship between the objects. Image captioning has many applications—for instance, as an aid developed to guide visually challenged people in travelling independently. This can be done by first converting the scenario into text and then transferring the text to voice messages. Image captioning can also be used in social media to automatically generate the caption for a posted image or to describe a video in real time. In addition, automatic captioning could improve the Google image search technique by converting the image into a caption and then using the keywords for further related searches. It can also be used in surveillance, by generating the relevant captions from CCTV cameras and raising alarms if any suspicious activity is detected [1].

2. Related Works

There exist numerous research works related to image captioning. Initially, image captioning was performed under constrained conditions. For example, Kojima et al. [2] used hierarchical actions to describe human activities from a video image, while Hede et al. [3] presented an image captioning method using a dictionary of objects and language templates. However, such constrained methods of image captioning are not applicable to daily life [4]. There are two other common types of image captioning methods: retrieval-based methods [5–10], and template-based methods [11–15]. In retrieval-based methods, the visually similar images are retrieved with their captions from the training dataset. Template-based methods require a predefined sentence template for each category of
images during training processes. Neither method is good enough, as the obtained image captions contain lack descriptive details. Today, encoder–decoder models are widely used for image captioning, as they can better produce the image captions accurately as compared with template- and retrieval-based models. In general, the encoder uses the convolutional neural network for feature extraction, while the decoder uses the recurrent neural network for caption generation. Moreover, an attention mechanism has been adopted to filter out the unnecessary information and, thus, improve the accuracy of image descriptions. Deng et al. [16] proposed an encoder–decoder-based image captioning model, where an adaptive attention mechanism and a long short-term memory network (LSTM) were used. Wang et al. [17] proposed a multilayer dense attention mechanism for image caption generation, using faster-RCNN as an encoder and LSTM-attend as a decoder. Zhang et al. [18] also presented an encoder–decoder model for remote sensing image captioning based on a convolutional neural network (CNN) and LSTM combined with a novel attention model called a visual aligning attention model. Gao et al. [19] proposed a hierarchical LSTM with an adaptive attention-based encoder–decoder model for visual captioning. Wang et al. [20] proposed an end-to-end deep learning approach for image captioning; essentially, a CNN and LSTM with attention were considered in the previous encoder–decoder structure for feature extraction and image captioning.

None of the aforementioned papers have discussed the color information of the objects. The object color is an important factor in object recognition and, thus, in image captioning. In the proposed method, color analysis is also addressed to make the image captioning more descriptive and accurate. Some of the research works based on color detection are discussed as follows: Ozturk et al. [21] used image segmentation and color analysis to distinguish fruits from leaves and branches. Liu et al. [22] proposed an algorithm to detect the color signal light image of a UAV. Zhang et al. [23] proposed an automatic grabbing robot using the OpenCV image processing library for object color recognition. Ashtari et al. [24] proposed a license plate recognition method based on color features, where the location of license plates can be recognized from their hue histogram and shape.

This paper proposes an ROS-based image captioning model, where the VGG16 convolutional network is employed as an encoder to extract the image features, and an LSTM neural network with attention is used as a decoder for semantic language processing. Moreover, Mask R-CNN with OpenCV is used for object detection and color recognition. In this paper, an enhanced image captioning model is proposed to produce captions with more descriptive information of images. The contributions of this paper are given as follows:

1. An enhanced image captioning algorithm is proposed that can successfully generate the textual description of an image;
2. The obtained results not only provide the overall information of the image, but also provide detailed explanation of a scenario showing the activity performed by each recognized object;
3. Color recognition of objects is addressed, such that more detailed information of an object can be identified. Thus, a more accurate caption can be generated;
4. The textual description of an image is displayed through a text-to-speech module that could provide more useful applications.

3. Methods

In this section, the methods used for the proposed image captioning are presented in detail. Figure 1 represents the overview of the processing model. The whole process can be divided into three parts: object detection, color analysis, and image captioning.
3.1. Object Detection

Object detection is related to computer vision and image processing, and deals with detecting objects from a certain class of images. The object detection methods can be divided into two categories: machine learning approaches, and deep learning approaches [25, 26]. In the deep learning approaches, RPN (Region Proposal Network) and SSD (Single Shot MultiBox Detector) are commonly used; RPN is based on the region proposals, and SSD is based on regression [27–30]. In this study, both the preliminary and full object detections were performed with the consideration of recognition efficiency and accuracy.

In preliminary object detection, it is preferable to screen out whether the input image contains the designated target objects. Recognition time-saving is the main concern at this stage. Thus, the one-stage learning method SSD neural network model is used. SSD can detect objects and recognize their positions at the same time. There are different versions of SSD, according to the CNN network used. In this study, SSD-MobileNet-V2 was adopted. SSD-MobileNet-V2 uses a depthwise separable convolution architecture to reduce the computational cost [31]. In SSD-MobileNet-V2, a new model is introduced with the inverted residual structure, where the nonlinearity in the bottleneck layer is removed [32]. There are two types of blocks in SSD-MobileNet-V2: one is the residual block with stride 1, and another is with stride 2, in which each block has 3 layers. The first layer is a 1 × 1 convolution with ReLu6, the second layer is the depthwise convolution, and the third layer is a further 1 × 1 convolution without any nonlinearity. The task of preliminary object detection is to identify whether the image contains the target objects. The consequent object recognition and feature extraction will be performed by a full object detection algorithm.

On the other hand, Mask R-CNN is used for full object detection. Mask R-CNN is a deep neural network that can solve instance segmentation problems in computer vision. In Mask R-CNN, a bilinear interpolation is used to obtain boundary information with small errors. This method, called ROIAlign, uses four boundary points to obtain the averaged pixel values of the center. Thus, the offset problem caused by traditional ROI pooling can be solved. Based on Mask R-CNN, the target area in the image is obtained for the target object. Then, the mask in the pixel level of the object is generated. After obtaining the candidate area through ROIAlign, a convolutional neural network is used to obtain the mask. The object contour obtained through foreground segmentation of the image is used for subsequent color analysis. In general, as the network deepens, the gradient explosion problem becomes more serious; it becomes difficult or even impossible for the network to converge. The deeper network brings another problem, in that the accuracy of the training set decreases as the network deepens.
3.2. Color Analysis

In order to obtain more details of the image, it is very important to detect and classify the object color in the image. In this study, the OpenCV computer vision library was used for color analysis. The basis of recognizing color is to extract the color channels and generate color gradients of color images. The color channels have different representations, such as RGB and HSV. RGB values represent the ratio of red, green, and blue in each channel, while the channel values in HSV refer to the hue, saturation, and value. Hue represents the basic properties of the color, saturation represents the purity of the color, and value represents the brightness of the color. RGB could produce different recognition results in different light intensities; the color information cannot be easily separated from luminance. On the other hand, the HSV representation is more likely to adapt to human visual characteristics [33,34]; here, each element of the color space can be separated, which makes the color recognition easier. From Figure 2, it can be seen that different color tones—such as red, orange, purple, etc.—have their own ranges of H values regardless of the saturation and brightness. Therefore, the object colors can be easily recognized by HSV values. Since the color format from OpenCV is in RGB, the image will be converted from RGB format to HSV format. In this study, by analyzing the HSV values of an image, the major color types of an object could be identified, as shown in Figure 3.

![Color recognition flowchart.](image)

Figure 2. The distribution of H values corresponding to different color tones, regardless of the S and V.

![Color recognition flowchart.](image)

Figure 3. Color recognition flowchart.

3.3. Image Captioning

The purpose of image captioning is to automatically describe the image with proper textual words. The challenge is to depict the visual relationship between objects with a suitable textual description. In this study, the process of image captioning was based on an encoder-decoder model. The encoder can extract the image features in a fixed-length vector and then decode that vector representation into a natural language description. Here, the encoder was a VGG16 convolutional neural network, and the decoder was an LSTM (long short-term memory) network with an attention mechanism. The architecture of the image captioning is shown in Figure 4. LSTM is an improved recurrent neural network, mainly used to solve the problem of gradient disappearance and gradient explosion during long sequence training. Long short-term memory can learn long-term dependence, and it is suitable for processing and predicting important events with long intervals and delays in time series. The attention mechanism in deep learning is essentially akin to the selective attention mechanism of humans. Human vision can quickly scan the image to identify the target areas that need to be focused on. Then, more attention can be paid to these areas in order to get detailed information about the targets. The inputs to the LSTM are from the
The overall implementation process is shown in Figure 6.

This paper adopts the NVIDIA Jetson Nano as the edge computing device. According to the hardware configurations, the NVIDIA Jetson Nano is not suitable for executing relatively complex neural network models. Thus, the SSD-MobileNet-V2 model is used for preliminary object detection. When the SSD-MobileNet-V2 model recognizes the target object, the current image will be converted into OpenCV format, and then the associated ROS message will be sent to Image Topic using ROS open-source packages. Then, the GPU host can subscribe to this image message and convert it back to the OpenCV image format, used as input to both the image captioning algorithm and the object recognition algorithm simultaneously. Both of these processing results are integrated together. The generated text results are sent back to Jetson Nano via ROS. Then, Jetson Nano sends the received text result to a text-to-speech algorithm, and produces the voice description through Bluetooth. The overall implementation process is shown in Figure 6.

**Figure 4.** Color recognition flowchart.

### 3.4. Implementation

In this section, the overall implementation details of the proposed model, along with how to train the network to achieve better results, are discussed. The system architecture of this study is shown in Figure 5, where TensorRT and TensorFlow are the frameworks in NVIDIA Jetson Nano and GPU computing host, respectively. NVIDIA Jetson Nano is mainly used for image capturing and preliminary identification. The NVIDIA Jetson Nano supports CUDA and CUDNN, and it is then performed as an edge computing device, while the GPU computing host is used as the main kernel for object detection, color recognition, and image captioning. The key features of the GPU computing host are an NVIDIA RTX 2060 6 G GPU, Intel 8600 CPU, and 16 G DDR4 memory. Both the NVIDIA Jetson Nano and the GPU host are installed with the ROS operating system to facilitate data communication and information exchange between devices. ROS is a distributed processing framework that enables executable files to be individually designed and combined for processing during execution. ROS provides services such as hardware abstraction and message transmission between the nodes. Using the ROS framework, multiple functions can be performed simultaneously without affecting one another. Again, multiple development languages—such as C++ and Python—can be used for integration, and the corresponding languages are used for different functions, which brings many advantages in development. ROS also provides many tested open-source development packages, such as interface packages and driver packages [35].

In this paper, the NVIDIA Jetson Nano was adopted as the edge computing device. According to the hardware configurations, the NVIDIA Jetson Nano is not suitable to execute relatively complex neural network models. Thus, the SSD-MobileNet-V2 model is used for preliminary object detection. When the SSD-MobileNet-V2 model recognizes the target object, the current image will be converted into OpenCV format, and then the associated ROS message will be sent to Image Topic using ROS open-source packages. Then, the GPU host can subscribe to this image message and convert it back to the OpenCV image format, used as input to both the image captioning algorithm and the object recognition algorithm simultaneously. Both of these processing results are integrated together. The generated text results are sent back to Jetson Nano via ROS. Then, Jetson Nano sends the received text result to a text-to-speech algorithm, and produces the voice description through Bluetooth. The overall implementation process is shown in Figure 6.
Figure 5. Proposed system architecture.

Figure 6. Process flow diagram of the proposed system.

4. Experiments and Results

It is necessary to confirm whether the learning algorithms are workable. Thus, each algorithm was verified before integrating them together, and the corresponding results are shown and discussed in the following subsections.

4.1. Model Training and Datasets

In this stage, two models are trained for object recognition according to the context of use. One model uses the MSCOCO dataset (2014 version), and the other model uses a self-made traffic signal dataset. MSCOCO is an open-source dataset with multiple object features, popularly used to train algorithms for different purposes. In the MSCOCO dataset, there are 1.5 million objects, belonging to 80 object recognition categories and 91 image captioning categories. Comparisons between three open-source datasets are shown in Table 1.
Table 1. Comparisons of open-source datasets.

| Dataset Category       | Image Quantity | BBox Quantity |
|------------------------|----------------|---------------|
| PASCAL VOC (07++12)    | 20             | 21,503        |
| MSCOCO (2014trainval)  | 80             | 123,287       |
| ImageNet Det (2017train)| 200            | 349,319       |

Although the number of images in the MSCOCO dataset is not the most, as shown in Table 1, the number of bounding boxes is far more than in other commonly used datasets, and the numbers of various object sizes—such as small, medium, and large—are evenly distributed in comparison to other datasets, as shown in Table 2. The image content provided in MSCOCO is closer to daily life scenes. For the second object recognition model, a self-made traffic light dataset is used. Compared to the open-source dataset, the self-made dataset needs more effort to collect and organize the images. However, the self-training model can expand the types of objects that the model can recognize. In practice, a total of 100 images are self-collected for traffic light recognition. The collected images are first labeled, and the training is carried out with 70% training and 30% verification. In general, if the image data are insufficient, the model recognition rate could be reduced. Data augmentation techniques can be applied to produce a greater variety of samples, such as flip, rotation, and scaling. In this work, original images are considered references, and these images are zoomed in and out by 0.5 times, and also rotated 45 degrees to the left or right. As a result of previous augmentation processes, the self-made dataset now has 500 images; the training process is shown in Figure 7. From Figure 8, it can be clearly seen that the recognition result is significantly improved after using the data augmentation technique. In Figure 8a, the trained model can detect just one traffic light using only original sample data. In Figure 8b, two traffic lights can be detected, as a result of the recognition model being trained with the augmented data.

Table 2. Distribution of object size (BBox size) in open-source datasets.

| Dataset Category       | Total   | Small  | Middle | Large   |
|------------------------|---------|--------|--------|---------|
| PASCAL VOC (07++12)    | 62,199  | 6983   | 19,677 | 35,539  |
| MSCOCO (2014trainval)  | 886,266 | 278,651| 311,999| 295,616 |
| ImageNet Det (2017train)| 478,806| 22,677 | 86,439 | 369,690 |

Figure 7. Flowchart of model training.
4.2. Preliminary Identification

In this step, preliminary screening of images is executed to identify the predefined target objects as required. If the target object is detected, then that image is sent to the next step for further processing; otherwise, it will simply be discarded. For example, two sets of images are taken as shown in Figure 9a,b, respectively. Each set contains three images, and different target objects are defined for each set; these images are used for preliminary identification. The recognized objects are framed as shown in Figure 10, where the target object in the first set is person, and the target object in the second set is traffic light. Therefore, from the preliminary image recognition step, the image that contains the target object can be screened out, as shown in Figure 11.

Figure 8. Recognition results from trained models: (a) without data augmentation; (b) with data augmentation.

Figure 9. (a) First set of images; (b) second set of images.

Figure 10. Object recognition results: (a) first set (target object: person); (b) second set (target object: traffic light).
In object recognition, the processing algorithm is subdivided into the steps of object segmentation and color analysis. When an image is given as an input to the object recognition algorithm, the algorithm will first recognize the object, including its range and position. Then, the algorithm extracts the color of each pixel from the segmented image and converts it into HSV values. After analyzing the color composition of objects, the main object colors can be obtained. The whole object recognition process with object segmentation and color recognition is shown in Figure 13. In addition, two images selected by the preliminary identification are fed to the image captioning and object recognition algorithms, and the output results are shown in Figure 14.

4.3. Image Captioning and Object Recognition

In this stage, the testing results of image captioning and object recognition are provided, where a GPU host is used to train the model. In the image captioning step, an input image is first fed to VGG16-no-FC, which is used to extract image features. These features become the inputs to an attention mechanism, and then a relative range of the focused part of the objects is extracted. Finally, descriptive sentences can be obtained through the LSTM network. The overview of this process is shown in Figure 12.

Figure 11. Results of preliminary identification: (a) first set; (b) second set.

Figure 12. Overview of the image captioning process.
object colors can be obtained. The whole object recognition process with object segmentation and color recognition is shown in Figure 13. In addition, two images selected by the preliminary identification are fed to the image captioning and object recognition algorithms, and the output results are shown in Figure 14.

Figure 13. Overview of the object recognition process.

Figure 14. Processing results: (a) image captioning; (b) object recognition.

4.4. Integration of Image Captioning and Object Recognition Results

Before performing the algorithm integration process, the results from the two algorithms need to be pre-processed. In the image captioning step, the output sentence needs to be segmented into individual words. Then, the linear search algorithm is used to search for the objects obtained from the object recognition algorithm in the list. If the object category appears in the image captioning output, the corresponding color will be inserted before the index of the object in the list. Finally, the list is reconstituted into a complete sentence according to the recognized objects and colors. The integration process of Figures 12–14 is illustrated in Table 3 and Figure 15. In order to make the caption understandable to visually challenged people, this paper uses the GTTS (Google Text-to-Speech) text-to-speech API. This includes three parts: sentence analysis, speech synthesis, and prosody generation; it produces the results with subtle sounds, such as lisps and accents. Compared with the speech synthesized by other speech synthesizers, it is more real and natural, and the gap with human performance is reduced by 70% [36].
Table 3. Distribution of object size (BBox size) in open-source datasets.

| Image Captioning results | A man sitting at a desk with a desktop. |
|--------------------------|----------------------------------------|
| Object recognition results | Person: blue  
Chair: blue  
Desk: white  
TV: black |
| Sentence segmentation | a, man, sitting, at, a, desk, with, a, desktop.  
a, man, sitting, at, a, white, desk, with, a, desktop. |
| Match text and insert color | a man in blue top sitting at a white desk with a desktop. |
| Integration results | a man in blue top sitting on a white bench with a black bicycle  
a busy street filled with lots of traffic and an orange traffic light. |

Figure 15. Integration results of Figure 13.

4.5. Enhanced Image Captioning Algorithm

From the algorithm integration process, we found that when the target object was not unique in an image, there could only be an ambiguous description. As shown in Figure 16, the textual description “two people standing in a kitchen preparing food” could not tell us who was actually wearing the red or orange top. In this paper, an improved method called the enhanced image caption algorithm is proposed, where ROIAlign is used to find the outlines of all objects, and then the PIL image processing suite (Python Imaging Library) is used to extract the objects individually. For example, originally, in Figure 16, there were two people in the image. The enhanced algorithm will generate two images, each containing just one person, where the other person is replaced with a black object, as shown in Figure 17. These two images are used as the inputs to the image captioning and color analysis algorithms. Viewing the results in Figure 17, it can be seen that there remain difficulties in specifically identifying each person. To further improve the proposed method, the extracted object can then be outlined by taking its maximum values from the top, bottom, left, and right—like a rectangle—as shown in Figure 18. After re-performing the image caption processing, the generated textual description with the original image is shown in Figure 19. It is clear that a more detailed and correct description can be obtained for the image with multiple similar objects.

Figure 16. Image captioning result of an image containing two similar objects.
4.6. Cases

Some more test images, including single objects and multiple objects, were used to verify the feasibility of the proposed image captioning scheme. The comparison between the traditional method [4,10,20] and the proposed method is illustrated in Table 4 and Figure 20. In Table 4, the sub-images 2 and 5 are taken as the examples for a single object and multiple objects, respectively. The main differences in the corresponding image captions are highlighted with underlines. More detailed explanations about the advantages of using the proposed method are illustrated in the following section. The images in Figure 20a are from the MSCOCO dataset, while the traffic light images in Figure 20b are from the self-made dataset. In Figure 20, the captions in black are the results of the trained model with CNN, LSTM, and attention. The captions in red are the results of the proposed model using the enhanced image captioning algorithm. It can be observed that the captions generated from the proposed model indeed improve the illustration quality by adding more semantic details. For examples, in sub-images 1–3, the red caption provides more color information than the black caption. Thus, with the proposed enhanced
model, the specification of objects is increased, which is helpful in image recognition for an explanation of surveillance camera footage. In sub-image 4, the caption describes the activity of the person, along with clothing color and object color. Sub-image 5 is an example of multiple similar objects; the black caption provides the information that there are two people sitting on a bench, but the red caption adds each individual activity, along with clothing information. Similarly, sub-image 6 is an example of two baseball players; the red caption indeed provides more details of each player in terms of color and activity. Moreover, in sub-images 7–12, the captions from the proposed enhanced model provide more color information of the traffic lights, such as red or green. In summary, the proposed image captioning model can generate textual descriptions more accurately in terms of color information and individual activity for each object. For a better illustration of the impact of the examples, some YouTube videos have been created to show the performing process: https://youtu.be/njR1NsXCFhs and https://youtu.be/weziAv3dLwg (accessed on 15 December 2021).

**Figure 20.** Examples of captions generated by the traditional image captioning algorithm (in black) and by the proposed model (in red). (a) First set of images; (b) second set of images.
Table 4. Comparison between the traditional method and the proposed method.

|                      | Single Object | Multiple Objects |
|----------------------|---------------|------------------|
| (Sub-images 2, 3, 7, 8, and 11 in Figure 20) | Captions in sub-image 2, e.g., a cow standing in a field | Captions in sub-image 5, e.g., a couple of men sitting on a bench, a man in blue top sitting on a bench and a man in green top reading a book |

5. Conclusions and Future Work

In this study, an encoder–decoder-based enhanced image captioning model is proposed. This model is applicable to images containing unique objects as well as more similar objects. The model can explain the scenario present in an image as well as adding color to the recognized object, which helps to provide better understanding of the scene. Furthermore, the color recognition adds more information to describe the traffic light signal, which is helpful in assisting visually challenged people. In the future, adding more data to the dataset could be considered in order to increase the recognition rate. To enhance the image captioning results, a generative adversarial network (GAN) could be used to fill in the background of the extracted object image in order to provide a more accurate description. Furthermore, it is worth paying more attention to improving the quality of life of visually challenged people, so that everyone can experience the benefits provided by deep learning.

Author Contributions: Conceptualization, Y.-H.C., Y.-J.C. and R.-H.H.; methodology, Y.-J.C. and R.-H.H.; software, Y.-J.C., R.-H.H. and Y.-T.Y.; validation, Y.-H.C. and Y.-T.Y.; writing—original draft preparation, Y.-H.C.; writing—review and editing, Y.-H.C. and Y.-T.Y.; supervision, Y.-H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ministry of Science and Technology, Taiwan, under Grant MOST 110-2221-E-182-055.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Image Captioning. Available online: https://www.slideshare.net/mz0502244226/image-captioning (accessed on 10 March 2021).
2. Kojima, A.; Tamura, T.; Fukunaga, K. Natural Language Description of Human Activities from Video Images Based on Concept Hierarchy of Actions. Int. J. Comput. Vis. 2002, 50, 171–184. [CrossRef]
3. Hede, P.; Moellic, P.; Bourgeois, J.; Joint, M.; Thomas, C. Automatic Generation of Natural Language Descriptions for Images. In Proceedings of the Recherche D’Information Assistée Par Ordinateur, Avignon, France, 26–28 April 2004; pp. 1–8.
4. Shuang, B.; Shan, A. A survey on automatic image caption generation. Neurocomputing 2018, 311, 291–304.
5. Ordonez, V.; Han, X.; Kuznetsova, P.; Kulkarni, G.; Mitchell, M.; Yamaguchi, K.; Stratos, K.; Goyal, A.; Dodge, J.; Mensch, A.; et al. Large-scale retrieval and generation of image descriptions. Int. J. Comput. Vis. 2016, 119, 46–59. [CrossRef]
6. Gupta, A.; Verma, Y.; Jawahar, C.V. Choosing linguistics over vision to describe images. In Proceedings of the AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012; pp. 606–612.
7. Farhadi, A.; Hejrati, M.; Sadeghi, M.A.; Young, P.; Rashitchian, C.; Hockenmaier, J.; Forsyth, D. Every Picture Tells a Story: Generating Sentences from Images. In Proceedings of the European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; pp. 15–29.
8. Ordonez, V.; Kulkarni, G.; Berg, T.I. Im2Text: Describing images using 1 million captioned photographs. In Proceedings of the 24th International Conference on Neural Information Processing Systems, Granada, Spain, 12–15 December 2011; pp. 1143–1151.
9. Kulkarni, G.; Premraj, V.; Dhar, S.; Li, S.; Choi, Y.; Berg, A.C.; Berg, T.I. Baby talk: Understanding and generating simple image descriptions. In Proceedings of the Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011; pp. 1601–1608.
