International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-11 Issue-4, November 2022

An Alternative Fashion to Automate the Appropriateness of ALT-Text using Microsoft Computer Vision API

Karamjeet Singh Gulati, Anupreet Sihra, Veena Khandelwal, Sergej Dogadov

Abstract. Designing and releasing of software's in production that contains images takes a lot of time due to the need of finding ALT-text attributes for the images embedded in the applications. This paper automates the task of writing ALT-text attributes in HTML, especially if image integration is large with the use of python PIP package and Microsoft Computer Vision API. This will save huge time and efforts of the developers by automating the task of captioning images manually up to a great extent. The challenge that confronts us is the quality of annotations generated by the machine with respect to the human generated annotations. To study the appropriateness of the captions delivered by APIs, a blend of human and machine assessment was used. We have noticed a high similarity in human and machine generated annotations as we obtained individual and cumulative BLEU score metric . Another metric is confidence score with a percentage mean of 0.5 .Also, we have calculated the time taken per caption which is 1.6 seconds per image which took 6.01 minutes to caption 200 images..

Keywords: ALT-text, Beautiful Soup, Image Captioning automation, Microsoft Computer Vision API, PIP Package.

I. INTRODUCTION

All developers want to design and release software's that goes to the production seamlessly, without defects and does not have rollback failures. One major problem faced by developers is that it takes a lot of time to find the ALT-text attributes for the images embedded in their applications. Alt text is a requirement for web accessibility. It is used to replace an image if it cannot be loaded. Another reason to implement alt text is to boost your site's SEO. By doing so, it will help improve your site's search rankings. If we ignore this problem, developers may miss critical deadlines as it is a tedious task to caption images manually. Also, there is no guarantee that the user will view all the image content of the browsed website. Image captioning [1] is the task of enunciating precise description of the visual content in images for converting sequence of images to sequence of words which lies at the convergence of computer vision [2] and natural language processing. Fig 1 demonstrates the image captioning with Microsoft Computer Vision API [3] where we have passed an image URL in JSON format and received a JSON file [3] which captioned the image as "***Christian Thompson in a suit and tie***" with a confidence score of 0.55 i.e., 55% confidence. All we need is to use the computer vision API service of Azure, get the API key and need to pass in the terminal with the HTML path.


```
{
  "description": {
    "tags": ["person", "suit", "man", "building", "necktie", "clothing", "outdoor",
"standing", "wearing", "business", "posing", "dressed"],
    "captions": [{
      "text": "Christian Thomsen in a suit and tie",
      "confidence": 0.5538171529769897
    }]
  },
  "requestId": "ff2d34d2-6e59-47ae-9223-3455e3814f91",
  "metadata": {
    "height": 480,
    "width": 720,
    "format": "Jpeg"
  },
  "modelVersion": "2021-05-01"
}
```


Fig. 1 (a) Embedded Image (b) Response received from Microsoft Computer Vision API

Other major advances include attention-based recurrent neural networks which are based on natural language processing [4].

Lately, the successful use of profound neural networks in computer vision and natural language processing tasks have motivated numerous AI specialists to investigate new research opportunities at the common point of these discrete domains [5]. We have not discovered many recognized studies on building open-source models for this promising technology. Thus, we fostered a profound vision model that recognizes a wide scope of visual content and addresses the arduous task of automatically captioning the information embedded in images into textual form with high accuracy. Computer Vision digs to computerise the work that the human visual system can do.

Manuscript received on 25 October 2022 | Revised Manuscript received on 31 October 2022 | Manuscript Accepted on 15 November 2022 | Manuscript published on 30 November 2022.

***** Correspondence Author (s)
Karamjeet Singh Gulati*, SRM Institute of Science & Technology, Delhi NCR Campus, Ghaziabad (U.P), India. E-Mail: karamjeetsinghgulati111@gmail.com
Anupreet Sihra, Banasthali University, Rajasthan, India. E-Mail: anupreetsihra5@gmail.com
Dr. Veena Khandelwal, SRM Institute of Science & Technology, Delhi NCR Campus, Ghaziabad (U.P), India. E-Mail: vn.khandelwal@gmail.com
Sergej Dogadov, Technische Universität, Berlin, Germany. E-Mail: s.dogadov@tu-berlin.de

Enhanced feature extraction and visual recognition has tremendously helped in recognizing multitudes of objects in images with little effort [5]. Natural language processing targets concept extraction from images to guarantee clear understanding of objects in images, to precisely interpret the embedded semantics. Moreover, encompassing mindfulness on a walkway is critical for a safe route, particularly for individuals who are visually impaired or outwardly disabled [6]. A reasonable and intelligent framework is vital for this reason [7]. Therefore, we intended to utilise our innovation to solve this problem of automatic captioning of images and produced an open-source model to solve the same. In this paper, we have used Microsoft Computer Vision API for analysis of images that highlights rich features found in an image and have built a python PIP package [8] which automates the task of inserting alternative text attributes for image tags. To distinguish image content and labels, this API utilises domain explicit models and descriptors. The approach used, uses a well-known python library called Beautiful Soup [9]. Beautiful Soup is a python library that is used for web scraping purposes to haul the information out of HTML and XML documents. With the help of this library, one can search the URLs with img tags in HTML files. If the file has some URL present in it, JSON converter sends a request to Microsoft Vision API and image captions can be extracted in string format which are then injected in the new HTML file that will contain the most accurate captioning of images. This will save enormous time and exertion of the developers by automating the task of captioning [8,10]. Human estimations can produce the fairest results for the generated captions quality [11]. However, they are labor-intensive and time-consuming. Thus, to evaluate the performance of the model, we can use a metric called the BLEU score [12,13]. BLEU stands for Bilingual Evaluation Understudy, which helps in evaluating a generated sentence to a reference sentence. BLEU score is prominent in evaluation because it was one of the initial strategies used in pre-programmed evaluation of machine interpreted text [14] and has a sensible relationship with human generated text. Also, we can get the implementation of BLEU score in the Python Natural Toolkit library or NLTK which can be used to consider generated text in contrast to a reference text. In this paper, we have utilised the BLEU score to make a cross reference between the captions generated by the local speaker and the machine in the wake of preparing the overview of information contained in images [15]. The accuracy level of the BLEU score lies between 0 and 1. Prediction of the model is more accurate if all the BLEU scores are high. For the estimation of BLEU score, NLTK allows us to determine the weighting of different n-grams. Thus, it provides us numerous metrics in BLEU to determine results in a more explicit manner than individual BLEU score and cumulative BLEU score. The evaluation of matching grams of a certain order is known as individual N-gram score, for example - single words (1-gram) or word pairs (2-gram or bigram) whereas the evaluation of individual n-gram scores at all orders from 1 to n and weighting them by computing the weighted geometric mean is known as cumulative score. Further, we have collected confidence scores [15,16] which is a parameter that helps to remunerate the developer's request. We have analysed that further in section 3. We attained a mean confidence score of 0.5 with the help of Microsoft Computer Vision API. We have collected the response time from API and calculated the average Latency [17] score which is 1.6 second per image i.e., for every 2 seconds, we are captioning three images. These measurements yield a specific score addressing the likeness between the original and machine generated captions. While there are several other potential ways to quantify the quality of generated captions, all the previously mentioned measurements, depend on either lexical or semantic data to measure the similitude between the machine and human generated captions for images.

II. MATERIALS AND METHODS

To approach this kind of automation we need to find some way of fostering an application for the developers, so that it can automatically generate captions whenever needed. Hence, we built a PIP package in python with the integration of Microsoft Computer Vision API [18] which helps in achieving automatic captioning with the call in the terminal. The API triggers and injects ALT attributes in the HTML file. Moreover, a reference of path to the HTML file and API key must be passed as parameter in the function.

A. Microsoft Computer Vision API

Microsoft offers many high-quality services tools and powerful AI services such as cognitive services which includes RESTful services that allow user-interaction globally on any device, it has a wide use of APIs that allow users for processing human language, sentiments, emotions, physical characteristics, audio and much more [19]. The Cognitive Service APIs are grouped into dissimilar categories each containing a trained set of APIs based on the requirements by the client. Some of the major services include Vision API, Knowledge API, Language API, Speech API. Vision API, is an image-processing API that is highly valuable in our image captioning model because it uses Natural Language Processing, then we have Knowledge API which helps in discovering information and area as per client's prerequisites. Furthermore, Language API helps in processing Natural Language whereas Speech API helps in empowering voice acknowledgement, check, and sound change into text. Microsoft also offers Bing web crawler that searches images, news and video according to client's requirements as shown in Table I.

Table 1. Categories of Microsoft Cognitive Services

Category	Description
Computer vision API	Provides image-processing algorithms
Language	Processes Natural Language
Search	Create a search-based service which reflects results from Bing in text, images or videos etc

These trained APIs under the Vision category include various APIs that perform divergent functions such as Computer Vision API provides image processing algorithms which are extremely useful in our image captioning model. Content Moderator helps in pre-determined monitoring of images, videos, and text automatically whereas Video API shows the spatial relationship between detected objects and images while identifying faces and other elements within videos.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Table 2. Vision API

API	Description
Computer vision API	Provides image-processing algorithms
Content Moderator	Helps in pre-determined monitoring of images, videos, and text automatically
Video API	Provides spatial relationship between detected objects and images
Video Indexer	Uses A.I platform for improved video insights
Face API	Create a search-based service which reflects results from Bing in text, images, videos etc
Emotion API	Detects people's emotions in an image based on face detection
Custom Vision Service	Helps in image processing while generating captions

Next, we have Video indexer, which utilises AI models that enhance video cooperation and concentrates on every bit of video to extract useful information from it so that the content is more discoverable. Face API is used for face detection and Emotion API detects people's emotions in an image based on face detection. Furthermore, in Custom Vision Services user trained labelled data can be used for image captioning as shown in Table II. Here, Microsoft Vision API can read the text from raw images to obtain desirable results or not. Before validation we must refer to this segment. In the next section of our study, we will discuss whether Microsoft Vision API proposed by Microsoft Azure has sufficiently powerful CNNs to identify text in images or not, as well as the benefits and drawbacks of using the Microsoft Azure Computer Vision API.

B. Architecture of Microsoft Computer Vision API

Computer Vision API is believed to be persistent in transforming information contained in images into words. This application is intended to "draw out the rich information from images and process it to label them by properly utilising pre-trained AI models" [18,19]. The API layer is constructed with the use of Azure Functions. These APIs empower the application to transfer images and retrieve information from Cosmos DB. At this point, when images are transferred through API calls, they are retained in the Blob storage area. When we add documents to the Blob storage area, it activates an Event Grid message which is conveyed to the Azure Function. For further analysis, Azure Function transmits a link to the newly moved file in the Computer Vision API for evaluation. After data analysis from the Computer Vision API, Azure Functions makes sure that the outcomes are stored in Cosmos DB accompanied by the image metadata as shown in Figure 2.

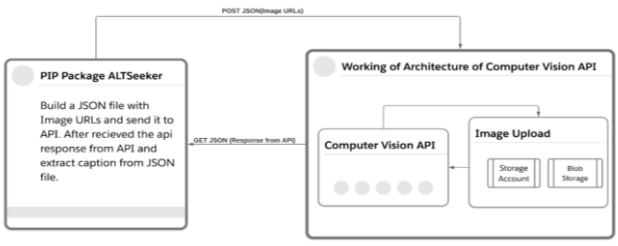

Fig 2. Architecture of Microsoft Computer Vision Cognitive Service

C. Need of Computer Vision API

Microsoft API provides cross-platform support where we can work on any language of our choice such as C#, java, python, Node.js etc. Microsoft API also provides pre-trained AI models which helps users to work easily by providing preliminary codes. The cloud helps to store data automatically. Since it is widely used, there are several tutorials available on the internet to help the developer to understand the structure and to establish, alter and inspect the code to get the most out of it.

D. Confidence score

API provides confidence score, a feature to see the similarity in generated and reference captions just like the BLEU score. The score varies from 0-100 where 0 denotes that no matching answer is found. If we have a higher score, it implies that there is more confidence in the answer. The score varies as per the image passed. If the score generated is between 90-100, that specifies an exact match. If the score is greater than 70, that shows HIGH confidence. If the score is between 50-70 that means MEDIUM confidence. If it is between 30-50, it means LOW confidence. If below 30 means VERY LOW confidence and 0 means no match. We can see our evaluation in section 4.

E. PIP (Preferred Installer Python)

As the developer contribution in open source is increasing and python has a large number of active supporting community of contributors, python allows their users to share and collaborate effectively. Python has a binary package installer called PIP where users can install packages and dependencies in their semi-isolated environment and can import the packages and save them in the machine locally. Google has TensorFlow which can be imported with PIP etc. We can install it to use for a particular application. So here, we have made a PIP package which can be installed by the developer to generate captions by passing an HTML file with Microsoft API key.

F. Implementation

To caption images automatically in a HTML file, we need to send image URLs to Microsoft Computer Vision API in json format [20] and inject received captions in the ALT-text attribute in HTML. For such a type of integration, we have made a pip package, named ALT Seeker [21], an open-source contribution. ALT Seeker - To generate captions for images automatically, we have developed a package installer in python better called PIP package which automates the task of inserting alternative text attributes for image tags in HTML. ALT Seeker here uses Microsoft CV API deep learning algorithm to caption images where it takes an HTML file as input, processes it, and generates a new html file with captions received from API and then it injects in ALT attributes in image tag. We have proposed an architecture for the same as shown in Figure 3.

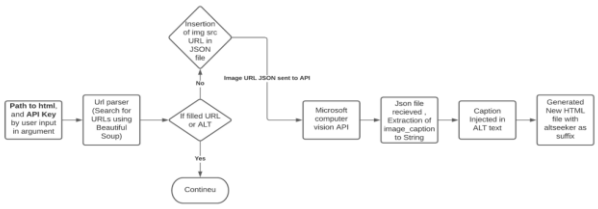

Fig 3. Architecture of ALT Seeker - PIP Package with Microsoft Computer Vision API integration to automate ALT-text in HTML.

Figure 3 shows the function created using the URL Parse library and returns whether there is any URL present in the html file or not. Then we have created a function for Microsoft Computer Vision API which takes an image URL and API key as argument inputs in the terminal, to send a post request to Microsoft Computer Vision API and a caption in string format for the image. Then we have made a function which takes the image source in string and converts it into a JSON format in order of URL received. If the image file is relative, we will find its absolute path by joining it to the absolute path of the HTML file. We have received the captioned_data. json file and we need to navigate to captioned text to get the required annotation of the image URL. We have used Beautiful Soup library for parsing and manipulating HTML DOMs for image tags which goes over each image tag and checks if its attribute has a value or not. If it does not carry a value, it will fill out the ALT with the corresponding caption of the image. It only fills out the ALT tags which do not have any value. We have used an argument parser in the terminal where the user needs to pass the HTML path along with the API key. At last, we have generated a new html file with the suffix “altseeker” at the end. If no name exists for the file, it will simply generate altseeker.html.

III. RESULT

After the successful application of the above approach, we can now calculate the evaluation metrics to compare our generated result with the real-life text written by the developer in ALT attributes of websites. We have gathered data of 200 images and 200 real written captions by developers taken as reference. We have also generated the ALT text of 200 images with ALT Seeker. Now, the developer's generated captions and the other containing machine generated captions in a dataframe [22]. To calculate the similarity level between the captions of both captions [19], we have calculated the BLEU score metric. For this, we have used the NLTK library and imported sentence_bleu. Further, we have also calculated 4 different Individual N-Gram Scores and Cumulative N-Gram BLEU scores. In Individual N-Gram, to calculate 1 gram weight, we need to specify the weight to one gram and 0 for others i.e., 1 for 1 gram and 0 for 2, 3 and 4 (1, 0, 0, 0) and we have plotted the same in Figure 4 whereas in Cumulative N-Gram Scores, to calculate 4 gram we have to equally distribute weight to 4 gram we need to mention 0.25 in each gram i.e., for 1,2,3 and 4, we can write (0.25, 0.25, 0.25, 0.25) and we have plotted the same as shown in figure 5. To analyse the graph, in figure 4, the mean of individual BLEU Score 1,BLEU Score 2,BLEU Score 3 and BLEU Score 4 are 0.30, 0.47, 0.64 and 0.70 respectively.

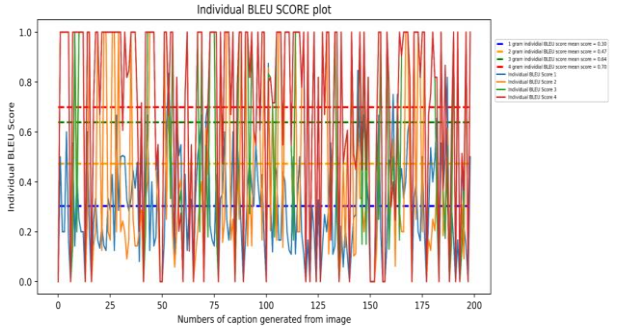

Fig 4. Plot of Individual BLEU 1,2,3 and 4 Score with a mean of 68.6

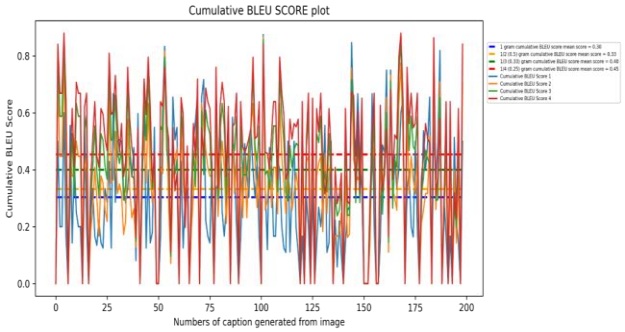

Fig 5. Plot of Cumulative BLEU 1,2,3 and 4 Score with a mean of 39.26.

In case of cumulative mean, we have calculated the mean of cumulative score, with a 1 gram distribution, the weights are (1,0,0,0), then ½ gram distribution as weights i.e, (0.5,0.5,0,0), for ⅓ gram distribution, we have taken weights as (0.33,0.33,0.33,0) and finally ¼ gram distribution where weights are (0.25,25,0.25,0.25) and we got the mean of 0.30, 0.33, 0.40 and 0.45 respectively. Hence, Our result shows high similarity between generated captions and developer written ALT-text

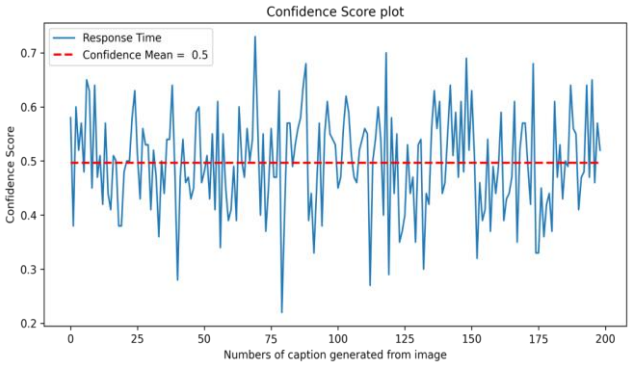

Fig 6. Plot of Confidence Score with a mean of 0.50.

Further, we can evaluate confidence scores generated in Microsoft API which is also a calculation metric which tells the level of similitude between human and model generated caption. Here, we have passed the raw string which we had stored earlier without any normalisation. All the above results clearly demonstrate that the text specified have high similarities and can be chosen for ALT-text automation. So, we have collected and stored all the generated confidence scores using Beautiful Soup which is visualised in Figure 6.

Retrieval Number: 100.1/ijrte.D73321111422
DOI: 10.35940/ijrte.D7332.1111422
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

```
<body>
<img class="1" src="https://i.postimg.cc/N0GbRbzc/0-00471900-1610095829-45.jpg" alt="">
<img class="3" src="https://i.postimg.cc/9fftD0DX/0d17d240-6375-4de4-8211-93ba3a0986ef.jpg" alt="">
<img class="4" src="https://i.postimg.cc/KvL9b1Md/0-55489800-1569394261-annapurna-mts-getty.jpg" alt="">
<img class="6" src="https://i.postimg.cc/76dV84YH/03-june-puppies.jpg" alt="">
<img class="7" src="https://i.postimg.cc/9fftD0DX/0d17d240-6375-4de4-8211-93ba3a0986ef.jpg" alt="">
    .
    .
    .
</body>
```

Fig 7. HTML file where ALT attributes are empty and passed in PIP Package as an argument.

Further,, we have acquired the mean for the confidence score and mean score is 0.5.. We have compared our captioned text and analysed it with human written content. In Figure 7, we can see a raw HTML file with empty ALT attributes. This empty HTML file has been parsed and sent to ALT Seeker which generates a new HTML file with captions injected in ALT-text as shown in Figure 8.

```
<body>
<img alt="a snake slithering on a rock" class="1" src="https://i.postimg.cc/N0GbRbzc/0-00471900-1610095829-45.jpg">
 <img alt="a river running through a forest" class="3" src="https://i.postimg.cc/9fftD0DX/0d17d240-6375-4de4-8211-93ba3a0986ef.jpg">
 <img alt = “a mountain with snow” class="4" src="https://i.postimg.cc/KvL9b1Md/0-55489800-1569394261-annapurna-mts-getty.jpg">
<img alt = “a puppy lying in the grass” class="6" src="https://i.postimg.cc/76dV84YH/03-june-puppies.jpg">
<img alt=”a river running through a canyon" class="7" src="https://i.postimg.cc/9fftD0DX/0d17d240-6375-4de4-8211-93ba3a0986ef.jpg">
    .
    .
    .
</body>
```

Fig 8. New generated HTML file with captions injected by ALT Seeker.

Thus, we also calculated the latency taken by every image and calculated the time taken by every image. This process will produce captions of 3 images in 2 seconds i.e., 1.6 second per image as we have the meantime in Figure 9 where it just took 6.01 minutes to caption all 200 images whereas if we try to do this manually it can take several hours to caption the same images manually.

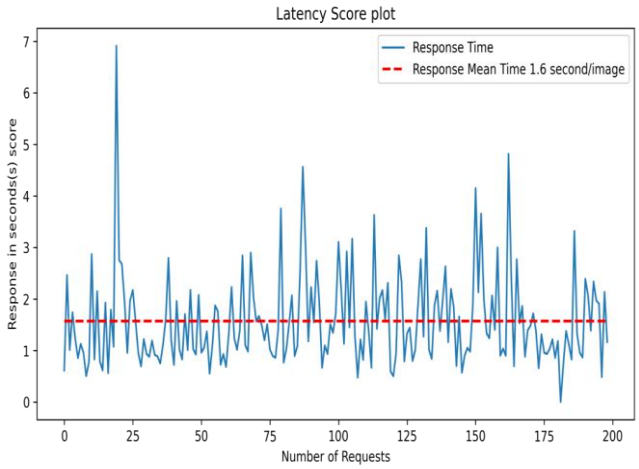

Fig 9. Plot of Latency Score plot with a mean of 1.6 second (s) per image

Thus, we can say that our ALT Seeker generated captions have high similarity to the developer's own written alt text and it's much faster than the traditional method.

IV. DISCUSSION

According to initial surveys CNNs and RNNs were hearty enough to generate image captions automatically [7]. To start with, many researchers extracted a significant level of elements from a CNN model and afterward utilised a RNN model to generate resulting captions of the images. While others embraced a multimodal structure in which necessary features of images are embedded in a multimodal network [8]. This multimodal network is then used to foresee the inscription of images word by word. But further evaluations and performance metrics demonstrated that the innovation is not yet fully grown enough to achieve very precise captioning of visual content [10]. Thus, this paper explores and enhances this mechanism to aid developers in automatic and faster translation of visual content into sequence of words with the help of Microsoft Computer Vision API. The major advantage of using Microsoft Computer Vision API over multimodal networks is that during the training of models, some image tags are randomly covered and the model is asked to identify the covered portions relative to the other parts of image. Even though the dataset utilised for fine-tuning just covers a little subset of the most widely recognized articles in the visual vocabulary, the pretrained model can sum up to any pictures that depict similar scenes with high accuracy.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Indeed, it is one of the best captions generating models that does not depend on caption explanations, empowering it to work with existing images developed for object identification. To survey the precision of the captions produced by APIs, a mix of human and machine evaluation was utilised [13]. Concerning the machine evaluation part of this model, BLEU score metric is used to survey the overall effectiveness of the API and the quality of the generated captions. BLEU score metric aided in calculating the accuracy about the captions produced by the model versus the captions produced by a human brain Also, the mean BLEU score obtained from Al seeker [21] was an average of 38.1 whereas we have obtained an individual mean of 68.6 and in case of cumulative BLEU Score, we got 39.26. Thus, our results indicate we have high similarity. As inferred from pymia [8], a PIP package used for medical image analysis, inspired us to develop our own PIP package with no code. Users must directly pass parameters as arguments in the terminal which is easy to use. Finally, after testing this model on an enormous number of human-produced captions portraying thousands of images, the model accomplished innovative results with generous improvement for objects it had not seen previously.

Owing to the fact that it provides pre-trained AI models, sometimes it takes time for the user to get familiar with the code. These cloud services are expensive to maintain and need stable internet access during data transmission as we are sending the data to a remote server every time we use the cloud. Transferring the data to other platforms or systems is not possible with Microsoft Azure. It has a lower degree of confidence, so sometimes it provides generic captions which give wrong predictions while validation. We can eliminate Azure services by choosing open-source models, or we can make our own models, which can be trained with a particular niche of images to make more in depth captions for a specific domain.

V. CONCLUSION

The experiment above shows that the ALT-text written by the developer can be automated with the help of an image captioning model. The evaluated results clearly proves that there is a similarity between human written and machine generated captions. This can be further included with the extension / Plugin of VS-Code, Sublime IDE, Atom IDE etc., which will save developer's time of generating ALT text. The architecture proposed in this paper can be used to automate any captioning task with cloud APIs provided by cloud services like we have used Microsoft Computer Vision API here.

REFERENCES

1. O. Vinyals, A. Toshev, S. Bengio and D. Erhan. Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 4, pp. 652-663, 1 April 2017, doi: 10.1109/TPAMI.2016.2587640. [CrossRef]
2. Kenneth Tran, Xiaodong He, Lei Zhang, Jian Sun, Cornelia Carapcea, Chris Thrasher, Chris Buehler, Chris Sienkiewicz. Rich Image Captioning in the Wild. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2016.
3. Martinez Gutierrez, Maria Fernanda. Automated Image Captioning: Exploring the Potential of Microsoft Computer Vision for English and Spanish. Université de Genève. Master, 2019. https://archive-ouverte.unige.ch/unige:132748.
4. Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan. Show and Tell: A Neural Image Caption Generator. Computer Vision and Pattern Recognition https://arxiv.org/abs/1411.4555.
5. Hasnine, Mohammad Nehal,Flanagan, Brendan, Akcapinar, Gokhan, Ogata, Hiroaki Mouri, Kousuke, Uosaki, Noriko. Distributed, Ambient and Pervasive Interactions" (LNCS, volume 11587) (2019):346-358. http://hdl.handle.net/ 2433/243253. [CrossRef]
6. F. Ahmed, M. S. Mahmud, R. Al-Fahad, S. Alam and M. Yeasin. Image Captioning for Ambient Awareness on a Sidewalk. 2018 1st International Conference on Data Intelligence and Security (ICDIS), 2018, pp. 85-91, doi: 10.1109/ICDIS.2018.00020. [CrossRef]
7. Michalik, Samuel. Deep learning and visualization of models for image captioning and multimodal translation. Praha, 2020. Bakalářská práce. Univerzita Karlova, Matematicko-fyzikální fakulta, Ústav formální a aplikované lingvistiky. Vedoucí práce Helcl, Jindřich. http://hdl.handle.net/20.500.11956/11937.
8. Alain Jungo, Olivier Scheidegger, Mauricio Reyes, Fabian Balsiger. pymia: A Python package for data handling and evaluation in deep learning-based medical image analysis. Computer Methods and Programs in Biomedicine,Volume 198, 2021. https://doi.org/10.1016/j.cmpb.2020.105796. [CrossRef]
9. Hajba G.L. (2018). Using Beautiful Soup. In: Website Scraping with Python. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-3925-4_3. [CrossRef]
10. Y. Bounab, M. Oussalah and A. Ferdenache. Reconciling Image Captioning and User's Comments for Urban Tourism. 2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA), 2020, pp. 1-6, doi: 10.1109/IPTA50016.2020.9286602. [CrossRef]
11. A. V. Potnis, R. C. Shinde and S. S. Durbha. Towards Natural Language Question Answering Over Earth Observation Linked Data Using Attention-Based Neural Machine Translation. IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, 2020, pp. 577-580, doi: 10.1109/IGARSS39084.2020.9323183. [CrossRef]
12. Naeha Sharif1, Lyndon White1, Mohammed Bennamoun1, and Syed Afaq Ali Shah .NNEval: Neural Network based Evaluation Metric for Image Captioning. https://openaccess.thecvf.com/content_ECCV_2018/papers/Naeha_Sharif_NNEval_Neural_Network_ECCV_2018_paper.pdf. [CrossRef]
13. Papineni, Kishore & Roukos, Salim & Ward, Todd & Zhu, Wei Jing. BLEU: A Method for Automatic Evaluation of Machine Translation. https://doi.org/10.3115/1073083.1073135. 4236. [CrossRef]
14. H Ahsan, N Bhalla, D Bhatt, K Shah. Multi-Modal Image Captioning for the Visually Impaired. arXiv preprint arXiv:2105.08106 [cs.CL], 2021 - arxiv.org. [CrossRef]
15. Fuhai Chen, Rongrong Ji, Jinsong Su, Yongjian Wu, and Yunsheng Wu. 2017. StructCap: Structured Semantic Embedding for Image Captioning. In Proceedings of the 25th ACM international conference on Multimedia (MM '17). Association for Computing Machinery, New York, NY, USA, 46–54. DOI:https://doi.org/10.1145/3123266.3123275. [CrossRef]
16. Yang Feng, Lin Ma, Wei Liu, Jiebo Luo. Unsupervised Image Captioning. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4125-4134. https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Nguyen_Deep_Neural_Networks_2015_CVPR_paper.pdf [CrossRef]
17. David Bermbach and Erik Wittern. 2016. Benchmarking Web API Quality. In Web Engineering, Springer International Publishing, Cham, 188–206. DOI:https://doi.org/10.1007/978-3-319-38791-8_11 [CrossRef]
18. Del Sole A. (2018). Getting Started with the Computer Vision API. In: Microsoft Computer Vision APIs Distilled. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-3342-9_2 [CrossRef]
19. Del Sole A. (2018). Introducing Microsoft Cognitive Services. In: Microsoft Computer Vision APIs Distilled. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-3342-9_1 [CrossRef]
20. Del Sole A. (2018) Invoking the Computer Vision API from C#. In: Microsoft Computer Vision APIs Distilled. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-3342-9_3 [CrossRef]

21. Altseeker - PIP Package for python for automating Alt text https://github.com/ksg98/altseeker
22. Dataset created and Evaluation implementation https://github.com/ksg98/Model-Evaluatiion-with-BLEU-Confidemce-and-Latency-with-dataset-usedyes

AUTHOR PROFILE

Karamjeeet Singh Gulati, Graduate from SRM University, currently working as a software engineer in Fidelity information Services (FIS) Bengaluru. Previously also worked in Unacademy, Walkover, Seekace, Hackveda as SDE and Mentor. Research Assistant in SRM CS Computer Network Lab specializes in Tracking Access Point exchange using wifi sniffers and Wireshark. My Specializations are ETL, Data Science, Machine Learning (Tensorflow and Keras) Backend technology (Node.js express).

Anupreet Sihra, Graduate From Banasthali University, India, Currently working in Barclays as Software development Engineer, Pune. Previously worked in Unacademy, Seekace as Mentor and Data Scientist. My Specializations are Data Science, Machine Learning and Deep Learning (Tensorflow and keras), Web scraping using Beautiful Soup, Backend Development in .Net framework C#, Node.js (Express,), Javascript, Java etc.

Dr. Veena Khandelwal, PhD in Computer Science and Engineering from Rajasthan Technical University, India M.Tech from University of Engineering Kota in Computer Science and Engineering. Presently working as Assistant Professor in SRM University, India. Currently having 30 years of experience. Specialization and certified in Cloud Computing from AWS. Publications: Amazon EC2 Spot Price Prediction using Regression Random Forests, Perceptive bidding strategy for Amazon EC2 spot instance market, Bidding Strategies for Amazon EC2 Spot Instances-A Comprehensive Review, Temporal and Spatial Trend Analysis of Cloud Spot Instance Pricing in Amazon EC2, Cost Optimization over Amazon EC2 Spot Instances, Improving Data Security and Availability Using Optimized Data Distribution in Multi-Cloud Storage,Secure and Efficient Data Storage in Multi-Clouds

Sergej Dogadov, Education: MSc and BSc in Computer Science from Technical University Berlin. Currently, writing his PhD thesis on the applications of mixture of Joint Energy Models and giving lectures on Python Programming for Machine Learning at the department of Intelligent Data Analysis and Machine Learning at Technical University Berlin. Publication includes: Variational Robust Subspace clustering and Automatic type identification of Alteration types in Historical manuscripts using Bayesian methods.