**Caption Generator for Images powered by Watson Visual Recognition**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **INTRODUCTION** | | | |  | | |  |  | |
|  | * 1. **Overview**   Image Captioning refers to the process of generating textual description from an image based on the objects and actions in the image. The ability to recognize image features and generate accurate, syntactically reasonable text descriptions is important for many tasks in computer vision. Some of the important applications of this project are self-driving cars and aid to the physically challenged people. | | | | | | | | | |
|  | * 1. **Purpose**   Automatic driving is one of the biggest challenges and if we can properly caption the scene around the car, it can give a boost to the self-driving system. Also, this project will be useful to create a product for the physically challenged people, which will guide them travelling on the roads without the support of anyone else. | | | | | | | | | |
| **2** | **LITERATURE SURVEY** | | | |  | | |  |  | |
|  | **2.1 Existing problem**  Image captioning [1-3] has been proposed that seek to ground the words in the predicted caption to regions in the image. As the visual attention is often derived from higher convolutional layers of a CNN, the spatial localization is limited and often not semantically meaningful. Researchers also handle this job by obtaining image features with an object detector, and generating captions through an attention LSTM.  Yao et al. [4] introduced two graph convolutional networks to find the semantic relationship graph, and a spatial relationship graph that classifies the relationship between two boxes into 11 classes, such as “inside”, “cover”, or “overlap”.  The self-critical reinforcement learning for sentence generation [5] has also proven to be important for state-of-the-art captioning approaches, such as those above. Liu et al. in [6] proposed an alternative reinforcement learning approach over a visual policy that, in effect, acts as an attention mechanism to combine features from the image regions provided by an object detector. The visual policy, however, does not utilize spatial information about these image regions.  Recent developments in NLP, namely the Transformer architecture [7] have led to significant performance improvements for various tasks such as translation, text generation, and language understanding. | | | | | | | | | |
|  | **2.2 Proposed solution**  This project aims to build an application which takes input as image, analyses it and generate the captions in the form of speech. To achieve this, this project uses IBM Services like node-red service to build a web UI where user uploads a picture. This picture is analyzed by visual recognition service and the analyzed description is then converted in to text and speech by text to speech service. | | | | | | | | | |
| **3** | **THEORITICAL ANALYSIS** | | | |  | | |  |  | |
|  | **3.1 Block diagram**  Generating a description of an image is called image captioning. Image captioning requires recognizing the important objects, their attributes, and their relationships in an image. It also needs to generate syntactically and semantically correct sentences. Fig. 1 shows the system design of image captioning.  **Fig. 1 System design of image captioning** | | | | | | | | | |
|  | **3.2 Software designing**  This project uses the following services: i) IBM Watson Visual Recognition Service The IBM Watson Visual Recognition service uses learning algorithms to analyze images for content such as objects, scenes, and faces. This service is used to create a visual recognition modeler which is used to automatically train a model to classify images for scenes, objects, or your custom content.  ii) Text to speech services  The Text to Speech service converts text to natural-sounding speech. This service streams the synthesized audio back with minimal delay. The audio uses appropriate cadence and intonation for its language and dialect to provide voices that are smooth and natural. The service can be used in applications such as voice-automated chatbots, as well as a variety of voice-driven and screenless applications, such as tools for the disabled or visually impaired, video narration and voice over, and educational and home-automation solutions.  iii) Node-red service  The node-red service provides a pre-configured Node-RED application, including a Cloudant service to store the application flow configuration. This is used to add services, generate and download the code, use the IBM Cloud Developer Tools CLI to run and debug locally, then deploy to Cloud Foundry or a DevOps Pipeline. | | | | | | | | | |
| **4** | **EXPERIMENTAL INVESTIGATIONS**  This project works the following two steps: (i) converting the scene into text and (ii) converting the text to voice. Both are now famous applications of Deep Learning. Initially, the visual recognition model generates class keywords that describe the image. Then, the text to speech service converts written text to natural-sounding speech. | | | | | | | | | |
| **5** | **FLOWCHART**  The Fig. 2 shows the node-red flow of the image caption generation.    **Fig. 2 Node-red flow of image captioning** | | | | | | | | | |
| **6** | **RESULT**  The Fig. 3 and Fig. 4 show the input and output screens.    **Fig. 3 Input screen**    **Fig. 4 Output screen** | | | | | | | | | |
| **7** | **ADVANTAGES & DISADVANTAGES**  The visual recognition model identifies and generates class keywords that describe the image. Then, the text to speech service converts generated text to natural-sounding speech. But, it is required to improve the performance of the system. | | | | | | | | | |
| **8** | **APPLICATIONS**   * Self-driving cars — Automatic driving is one of the biggest challenges, and if it can properly caption the scene around the car, it will give a boost to the self-driving system. * Aid to the blind — This motivated to create a product for the blind which will guide them travelling on the roads without the support of anyone else.   The above two applications can be done by first converting the scene into text and then the text to voice. Both are now famous applications of Deep Learning. | | | | | | | | | |
| **9** | **CONCLUSION**  The project caption generation for images powered by Watson visual recognition converts the scene into the text using visual recognition service, and the generated text to speech by Watson text to speech service. | | | | | | | | | |
| **10** | **FUTURE SCOPE**  CCTV cameras are everywhere today, but along with viewing the world, if we can also generate relevant captions, then we can raise alarms as soon as there is some malicious activity going on somewhere. This could probably help reduce some crime and/or accidents. | | | | | | | | | |
| **11** | **BIBILOGRAPHY**   1. K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In International conference on machine learning, pages 2048–2057, 2015. 2. Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4651–4659, 2016. 3. Z. Yang, Y. Yuan, Y. Wu, W. W. Cohen, and R. R. Salakhutdinov. Review networks for caption generation. In Advances in Neural Information Processing Systems, pages 2361–2369, 2016. 4. T. Yao, Y. Pan, Y. Li, and T. Mei. Exploring visual relationship for image captioning. In European Conference on Computer Vision, pages 684–699, 2018. 5. S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7008–7024, 2017. 6. D. Liu, Z.-J. Zha, H. Zhang, Y. Zhang, and F. Wu. Context-aware visual policy network for sequence-level image captioning. In Proceedings of the 26th ACM International Conference on Multimedia, MM ’18, pages 1416–1424. ACM, 2018. 7. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008, 2017. | | | | | | | | | |
|  | **APPENDIX** | | | |  | | |  |  | |
|  | | 1. **Source code**   <!DOCTYPE html>  <html lang="en">  <head>  <title>Visual Recognition</title>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>  <style>  .bg-light {  background-color: #00838F!important;  }  h4 {  color: #fff;  }  .custom-file {  margin-bottom: 14px;  }  .table .thead-dark th {  color: #fff;  background-color: #307d76;  border-color: #307d76;  }  h5{  text-align: center;  color: #bd6666;  }  .text-center {  text-align: center;  }  .imgdiv  {  align:center;  }  </style>  </head>  <body>  <nav class="navbar navbar-expand-sm bg-light">  <div class="justify-content-center">  <h4 class="text-center">Object Detection using Visual Recognition </h4>  </div>  </nav>  <br><br>  <div class="containetr">  <div class="row">  <div class="col-sm-2">  </div>  <div class="col-sm-8">  <div class="custom-file">  <input type="file" name="pic" accept="image/\*" onchange="readURL(this);" class="custom-file-input" id="customFile">  <label class="custom-file-label" for="customFile">Choose file</label>  </div>  <br>  <div class="imgdiv">  <img src="#" id="blah" class="rounded" alt="Selected Image">  </div>  <!--<md-button ng-click="send({payload:action()})">  Predict  </md-button>-->  <button type="submit" ng-click="send({payload:action()})" class="btn btn-success">Submit</button>  </div>  <div class="col-sm-2">  </div>  </div>  <div class="row">  <div class="col-sm-2">  </div>  <div class="col-sm-8">  <h5> The Recognised Objects </h5>  <table class="table">  <thead class="thead-dark">  <tr>  <th>Class</th>  <th>Score</th>  </tr>  </thead>  <tbody id="scoretable">  </tbody>  </table>  </div>  <div class="col-sm-2">  </div>  </div>  </body>  </html>  <script>  var x="";  function readURL(input) {  if (input.files && input.files[0]) {  var reader = new FileReader();  reader.onload = function (e) {  $('#blah')  .attr('src', e.target.result)  .width(150)  .height(200);  };  reader.readAsDataURL(input.files[0]);  x= input.files[0]  }  }  function getdata(data)  {  var html = '';  var classes='';  if(data != 0)  {  $.each(data, function(i){  var row = data[i];  console.log(row);  if(row.score>0.6)  {  classes +=row.class;  classes+=', '  html += '<tr>';  html += '<td>';  html += row.class;  html += '</td>';  html += '<td>';  html += row.score;  html += '</td>';  html += '</tr>';  localStorage.setItem("message",classes);  }  });  }  else  {  html+="<div>No Data</div>";  classes+= "<div>No Data</div>";  }  $('#scoretable').html(html);  }  (function(scope) {  scope.$watch('msg.payload', function(data) {  console.log('Position 2');  console.dir(data);  getdata(data);  });  })(scope);  // or overwrite value in your callback function ...  this.scope.action = function() { return x; }  </script> | | | | | | | |
|  | |  |  |  | |  |  | | |