Contents

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence. 2

BLEU Bilingual Evaluation Understudy. 4

CNN Convolutional neural networks. 2

GCP Google Cloud Platform. 2

GPU Graphical Processing Unit. 10

GUI Graphical User Interface. 11

LSTM Long short term Memory. 6

MLP Multi-layer perceptron. 26

MOP-CNN multi scale orderless pooling CNN. 4

NLP Natural Language Processing. 13

V2L Vision-to-Language. 5

VQA Visual Question Answering. 1

Chapter 1

Introduction

Visual Question AnsweringVQA involves an image and a related text question, to which the machine must determine the correct answer. This task spans the fields of computer vision and natural language processing, since it requires both the comprehension of the question and parsing the visual elements of the image. VQA is a practical setting to evaluate deep visual understanding,itself considered the overarching goal of the field of computer vision. Deep visual understanding can be defined as the ability of algorithm to extract high-level information from images and to perform reasoning based on that information. In this regard, VQA is an alternative to other tasks proposed to evaluate this capability. Examples include the visual Turing test , the task of image captioning, and recent works on visual dialogues.

Language and vision problems such as image captioning and visual question answering have gained popularity in recent years as the computer vision research community is progressing beyond "bucketed" recognition and towards solving multi-modal problems.The complex com positional structure of language makes problems at the intersection of vision and language challenging. But recent works have pointed out that language also provides a strong prior that can result in good superficial performance, without the underlying models truly understanding the visual content

The recent interest in VQA originates from the latest advances in computer vision on low- and mid-level tasks. This encouraged further research on higher-level tasks, and the combination of vision with other modalities,particularly language. Historically, one of the earliest integration's of computer vision with language was the SHRDLU system dating back to 1972 , which allowed the use of language to instruct a computer to move objects in a simulated "blocks world." Other attempts at creating conversational robotic agents were also grounded in the visual world. However, these early works were often limited to specific domains and/or simple language. Deep learning has now been applied to virtually every problem imaginable

in computer vision, and convolutional neural networks(CNN are approaching human performance in tasks such as image segmentation or object recognition. The success of deep learning on perceptual tasks drove an increasing enthusiasm for high-level tasks. VQA particularly embodies this confidence in achieving high-level image understanding.

An instance of VQA consists of an image and a related question given in plain text (see examples in Figure 1). The task for the machine is to determine the correct answer, which is, in current data sets, typically a few words or a short phrase.Two practical variants are usually considered, an open-ended and a multiple-choice setting. In the latter, a set of candidate answers are proposed. This makes the evaluation of a generated answer easier than in the open-ended setting,where the comparison between the machine's output and a ground truth (i.e., human provided) answer faces issues with synonyms and paraphrasing.

The task of VQA is a significant step toward general AI and a departure from low and mid-level tasks in classical computer vision. It requires relating visual concepts with elements of language, common-sense, and general knowledge.

1.1 Objectives

- To Collect data set of image and questions related to image: Dataset related to VQA is already available online. To use this available dataset in this project
- To Pre-process the image data and text data : Images obtained from data set must be pre-processed and also text Data should be tokenized and a vocabulary must be build before building the model
- To Build a functional visual questioning model which takes input as image and question: In order to get desired results , several different neural network with different parameters has to be tried before finalising the model.This project uses Keras and Tensorflow to develop the deeplearning model.
- To Develop web application which hosts this model: To develop web application to let other users use this vqa model.Here fast-api(python based web framework) is used to develop and host this model on web.
- To Deploy this web application on the internet: This project has been deployed on heroku and GCP ,so that others can access this model on internet.

Chapter 2

Literature Survey

This chapter includes Previous research where projects similar to the proposed system has been discussed.

2.1 Previous Research

In VQA, the image and the question are processed independently to obtain separate vector representations. Different methods for doing this are detailed and comprehended in the previous two sections. Now, in the next step of VQA, these features are mapped to a joint space, then combined and fed to answer generation stage. This literature review extensively identified wide assortments of techniques utilized for consolidating image and question features ranging from simple concatenation to complex joint attention networks.

2.1.1 VQA: Visual Question Answering:

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu ,Margaret Mitchell, Dhruv Batra,C. Lawrence Zitnick,Devi Parikh,Virginia Tech ,Microsoft Research

This paper introduces the task of Visual Question Answering (VQA). Given an image and an open-ended, natural language question about the image, the task is to provide an accurate natural language answer. It provides a dataset containing over 250K images, 760K questions, and around 10M answers. They set up an evaluation server and organized an annual challenge and an associated workshop to facilitate systematic progress. It demonstrates the wide variety of questions and answers in their dataset, as well as the diverse set of AI capabilities in computer vision, natural language processing, and commonsense reasoning required to answer these questions accurately. The questions solicited from human subjects were open-ended and not task-specific. For some application domains, it would be useful to collect task-specific questions. For instance, questions may be gathered from subjects who are visually impaired, or the questions could focus on one specific domain (say sports). This paper also mentions that Bigham created an application

that allows the visually impaired to capture images and ask open-ended questions that are answered by human subjects. Interestingly, these questions can rarely be answered using generic captions. Training on task-specific datasets may help enable practical VQA applications. It believes VQA has the distinctive advantage of pushing the frontiers on "AI-complete" problems, while being amenable to automatic evaluation. Given the recent progress in the community, they believe the time is ripe to take on such an endeavor.

2.1.2 Multi-scale Orderless Pooling of Deep Convolutional Activation Features:

Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik

Deep convolutional neural networks (CNN) have shown their promise as a universal representation for recognition. However, global CNN activations lack geometric invariance, which limits their robustness for classification and matching of highly variable scenes. To improve the invariance of CNN activations without degrading their discriminative power, this paper presents a simple but effective scheme called multi scale orderless pooling (MOP-CNN). This scheme extracts CNN activations for local patches at multiple scale levels, performs orderless VLAD pooling of these activations at each level separately, and concatenates the result. The resulting MOP-CNN representation can be used as a generic feature for either supervised or unsupervised recognition tasks, from image classification to instance-level retrieval; it consistently outperforms global CNN activations without requiring any joint training of prediction layers for a particular target dataset. In absolute terms, it achieves state-of-the-art results on the challenging SUN397 and MIT Indoor Scenes classification datasets, and competitive results on ILSVRC2012/2013 classification and INRIA Holidays retrieval datasets.

2.1.3 From Captions to Visual Concepts and Back:

Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K. Srivastava, Li Deng Piotr Dollar, Jianfeng Gao, Xiaodong He, Margaret Mitchell ,John C. Platt, C. Lawrence Zitnick , Geoffrey Zweig

This paper presents a novel approach for automatically generating image descriptions: visual detectors, language models, and multimodal similarity models learnt directly from a dataset of image captions. They use multiple instance learning to train visual detectors for words that commonly occur in captions, including many different parts of speech such as nouns, verbs, and adjectives. The word detector outputs serve as conditional inputs to a maximum-entropy language model. The language model learns from a set of over 400,000 image descriptions to capture the statistics of word usage. they capture global semantics by re-ranking caption candidates using sentence-level features and a deep multimodal similarity model. Their system is state-of-the-art on the official Microsoft COCO benchmark, producing a BLEU-4 score of 29.1

2.1.4 Show and Tell: A Neural Image Caption Generator

Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan

Automatically describing the content of an image is a fundamental problem in artificial intelligence that connects computer vision and natural language processing. This paper presents a generative model based on a deep recurrent architecture that combines recent advances in computer vision and machine translation and that can be used to generate natural sentences describing an image. The model is trained to maximize the likelihood of the target description sentence given the training image. Experiments on several datasets show the accuracy of the model and the fluency of the language it learns solely from image descriptions. Their model is often quite accurate, which was verified both qualitatively and quantitatively. While the state-of-the-art BLEU score (the higher the better) on the Pascal dataset is 25, their approach yields 59, compared to human performance around 69. This paper also shows BLEU score improvements on Flickr30k, from 56 to 66, and on SBU, from 19 to 28. Lastly, on the newly released COCO dataset, they achieved a BLEU-4 of 27.7.

2.1.5 What Value Do Explicit High Level Concepts Have in Vision to Language Problems?:

Qi Wu, Chunhua Shen, Lingqiao Liu, Anthony Dick, Anton van den Hengel

Much recent progress in Vision-to-Language (V2L) problems has been achieved through a combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). This approach does not explicitly represent high-level semantic concepts, but rather seeks to progress directly from image features to text. This paper investigates whether this direct approach succeeds due to, or despite, the fact that it avoids the explicit representation of high-level information. They proposed a method of incorporating high-level concepts into the successful CNN-RNN approach, and show that it achieves a significant improvement on the state-of-the-art in both image captioning and visual question answering. They also show that the same mechanism can be used to introduce external semantic information and that doing so further improves performance. They achieved the best reported results on both image captioning and VQA on several benchmark datasets, and provided an analysis of the value of explicit high-level concepts in V2L problems.

2.1.6 Visual7W: Grounded Question Answering in Images:

Yuke Zhu, Oliver Groth, Michael Bernstein, Li Fei-Fei

We have seen great progress in basic perceptual tasks such as object recognition and detection. However, AI models still fail to match humans in high-level vision tasks due to the lack of capacities for deeper reasoning. Recently the new task of visual question answering (QA) has been proposed to evaluate a model's

capacity for deep image understanding. Previous works have established a loose, global association between QA sentences and images. However, many questions and answers, in practice, relate to local regions in the images. This paper proposes to leverage the visually grounded 7W questions to facilitate a deeper understanding of images beyond recognizing objects. Previous visual QA works lack a tight semantic link between textual descriptions and image regions. Here they link the object mentioned to their bounding boxes in the images. Object grounding allows to resolve coreference ambiguity, understand object distributions, and evaluate on a new type of visually grounded QA. They proposed an attention-based LSTM model to achieve state-of-the-art performance on the QA tasks. Future research directions include exploring ways of utilizing common sense knowledge to improve the model's performance on QA tasks that require complex reasoning.

2.1.7 Making the v in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering:

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, Devi Parikh

Problems at the intersection of vision and language are of significant importance both as challenging research questions and for the rich set of applications they enable. However, inherent structure in our world and bias in our language tend to be a simpler signal for learning than visual modalities, resulting in models that ignore visual information, leading to an inflated sense of their capability. This paper proposes to counter these language priors for the task of Visual Question Answering (VQA) and make vision (the V in VQA) matter! Specifically, by balancing the popular VQA dataset (Antol et al., ICCV 2015) by collecting complementary images such that every question in their balanced dataset is associated with not just a single image, but rather a pair of similar images that result in two different answers to the question.Their dataset is by construction more balanced than the original VQA dataset and has approximately twice the number of image-question pairs and their complete balanced dataset is publicly available at http://visualqa.org/ as part of the 2nd iteration of the Visual Question Answering Dataset and Challenge (VQA v2.0). Further they benchmark a number of state-of-art VQA models on balanced dataset. All models perform significantly worse on balanced dataset, suggesting that these models have indeed learned to exploit language priors. This finding provides the first concrete empirical evidence for what seems to be a qualitative sense among practitioners. Finally, data collection protocol for identifying complementary images enabled to develop a novel interpretable model, which in addition to providing an answer to the given (image, question) pair, also provides a counter-example based explanation. Specifically, it identifies an image that is similar to the original image, but it believes has a different answer to the same question. This can help in building trust for machines among their users.

2.2 Summary of Literature Review

Visual Question Answering is an interesting combination of Computer Vision and Natural Language Processing which is growing by utilizing the power of deep learning methods.It is very challenging task where requires solving many subtasks like object detection, activity recognition, spatial relationships between objects, commonsense reasoning etc. VQA, now accepted as an AI-complete task, would be an essential step towards the AI dream of visual dialogue.This section critically examines the existing solutions presented all comparisons of similar methodologies in user-friendly manner, which will help the entire range of VQA researchers from beginners to experts. The identified limitations of current practices and also the promising research directions revealed in this paper will lead to great success of VQA and also its descendants, video question answering and visual dialogue. The main obstacle in the journey of VQA models towards AI dream is that it is not clear what the source of improvement.

Chapter 3

System Architecture and Methodology

This section defines block diagram of visual question answering used in this project. Methodology of the project is also explained in this section.

3.1 Block diagram

The Block Diagram for the proposed model consists of these blocks as shown in Fig 3.1.Visual Representation, Textual Representation, Merge and Predict answer. Visual Representation includes extracting the visual data from an image dataset and building a model with Convolution Neural Network(CNN). Textual Representation includes extracting the text data, building vocabulary and tokenizing the text, Encoding the tokenized text, and building a model with Word/Sentence Embedding and Long Short Term Memory (LSTM). Merge block includes Merging two Neural Network Models and Predicting the Answer.

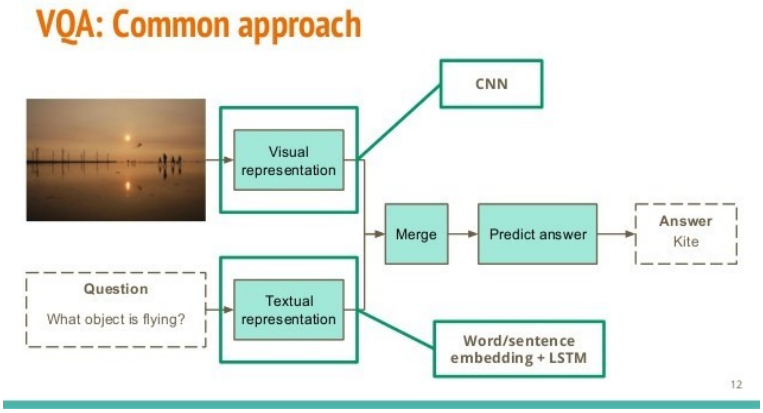

Figure 3.1: Block Diagram of VQA

3.2 Methodology

For this model the following design is applied , question is analyzed with a semantic parser with the aim of determining the basic computational units that are needed. This is the mapping from questions to layouts, which specifies both the modules needed to answer the question and the connections between them. Fig 3.2 gives an example of layout designed by this model. The final model combines the result from neural module network with a LSTM question encoder. In addition to the neural module network, this paper also collected a synthetic dataset, SHAPES, which includes different shapes in various positions. Fig 3.3 shows an example of an image in this dataset, and a layout to answer a particular question

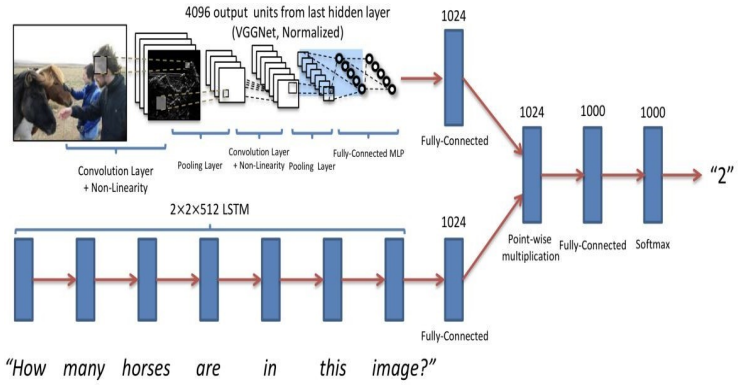

Figure 3.2: Proposed Architecture

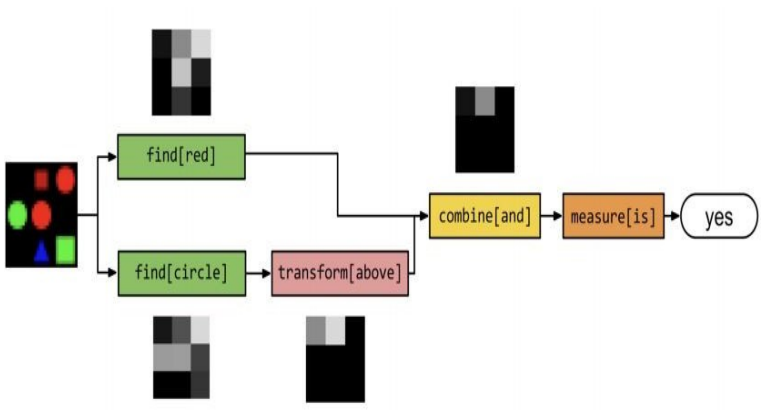

Figure 3.3: An example of an image in SHAPES dataset, and a layout to answer Is there a red shape above a circle

Chapter 4

Hardware and Software Components

This section will define hardware and software components used in this project.There are only few hardware requirements like GPU. Since this is a deeplearning project a lot of requirements are software components.

4.1 Hardware requirements

This Project is mainly focused building the model , There is no scope for hardware in this project. However the only hardware used in this project is Graphical Processing Unit.

4.1.1 Graphical Processing Unit

GPUs are optimized for training artificial intelligence and deep learning models as they can process multiple computations simultaneously. They have a large number of cores, which allows for better computation of multiple parallel processes. Additionally, computations in deep learning need to handle huge amounts of data.This makes a GPU's memory bandwidth most suitable.In this project, Nvidia Geforce GTX 1060 Ti GPU is used which has 4GB memory with GDDR5 Technology. The Figure 4.1 shows the comparison of bandwidth of CPU's and GPU's over the last two decades.This Graph shows that the Bandwidth of GPU's has exponentially grown when compared to CPU's.In this Graph slope of the Graph of GPU's is very much larger than CPU's.

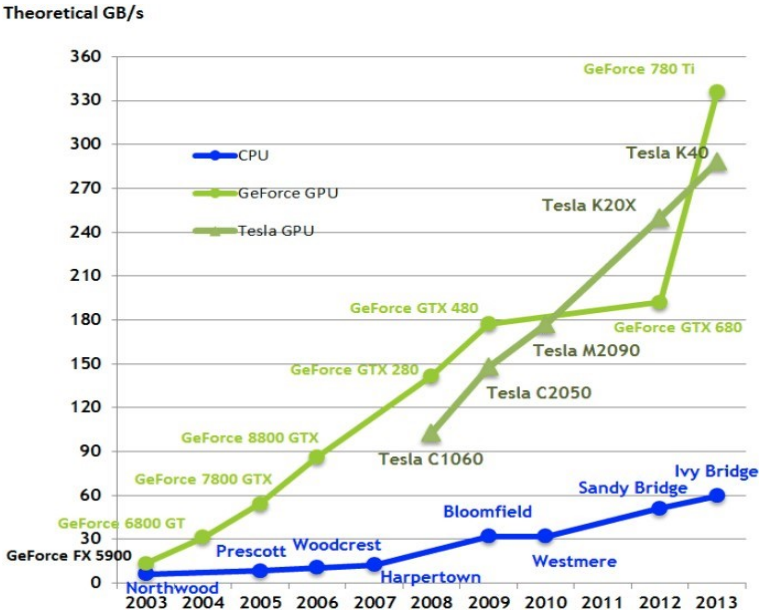

Figure 4.1: Comparison of bandwidth for CPUs and GPUs over time

4.2 Software Components

The Software Requirements of this Project are listed below. This project is purely focused on Software.

4.2.1 Google Cloud Run

Cloud Run is a managed compute platform that enables you to run containers that are invocable via requests or events. Cloud Run is serverless: it abstracts away all infrastructure management.

The logic inside the container has to be stateless, and you must specify a combination of memory vs CPU resources as well as allowed concurrencies. The cloud container will delete data that is stored on it. A persistent volume must be attached, or other data transfer methods used while the container is running, to prevent any stored data from being deleted. Afterwards, Cloud Run automatically adds or removes servers to meet user demand, only charging for what is used to the nearest tenth of a second. Cloud Run provides the user the http endpoint, container usage information, and control over the container's billing. The Cloud Run usage can be monitored easily in the GUI Metrics, along with logs from the container. Container deployment also has a built-in versioning feature, so a rollback to a previous container version is easy.

Google Cloud Run, which includes provisioning, configuring, scaling, and management, is targeting developers that want the ease of serverless with the portability from containers. Cloud Run will be supported by Datadog, NodeSource, GitLab, and StackBlitz.The Figure 4.2 shows the Google cloud products and how they are used to develop and deploy modern applications in cloud.

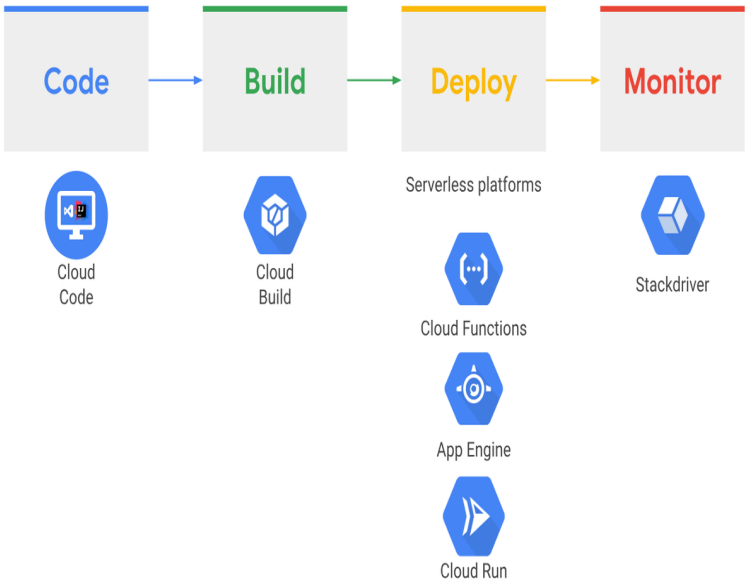

Figure 4.2: Google's serverless stack

4.2.2 Docker

Docker is a set of platform as a service () products that use OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels. Because all of the containers share the services of a single operating system kernel, they use fewer resources than virtual machines.

Docker provides the ability to package and run an application in a loosely isolated environment called a container. The isolation and security allow you to run many containers simultaneously on a given host. Containers are lightweight and contain everything needed to run the application, so you do not need to rely on what is currently installed on the host. You can easily share containers while you work, and be sure that everyone you share with gets the same container that works in the same way.

Docker uses a client-server architecture. The Docker client talks to the Docker daemon, which does the heavy lifting of building, running, and distributing your Docker containers. The Docker client and daemon can run on the same system, or you can connect a Docker client to a remote Docker daemon. The Docker client and daemon communicate using a REST API, over UNIX sockets or a network interface. Another Docker client is Docker Compose, that lets you work with applications consisting of a set of containers.The Figure 4.3 Shows the Docker architecture.Here the docker client uses the Registry files and Images to develop the Container. The Docker daemon is the heart of this architecture, this daemon communicates with docker client and recives command to build, run and pull containers.

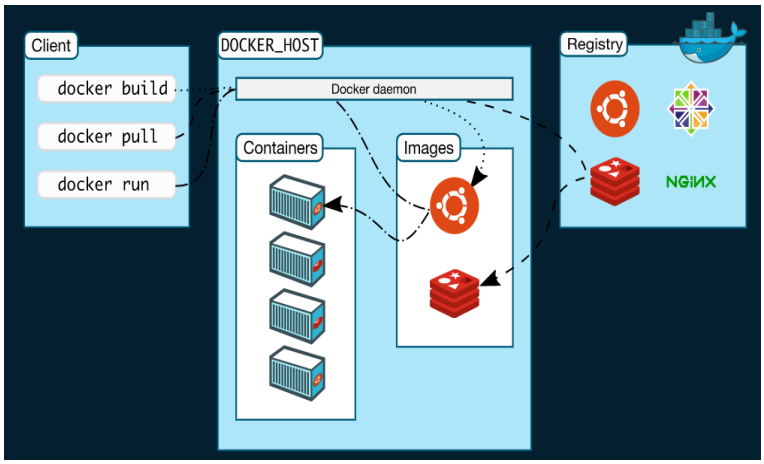

Figure 4.3: Docker architecture

4.2.3 SpaCy

spaCy is a free, open-source library for NLP in Python. It's written in Cython and is designed to build information extraction or natural language understanding systems. It's built for production use and provides a concise and user-friendly API.

While some of spaCy's features work independently, others require trained pipelines to be loaded, which enable spaCy to predict linguistic annotations – for example, whether a word is a verb or a noun. A trained pipeline can consist of multiple components that use a statistical model trained on labeled data. spaCy currently offers trained pipelines for a variety of languages, which can be installed as individual Python modules. Pipeline packages can differ in size, speed, memory usage, accuracy and the data they include. The package you choose always depends on your use case and the texts you're working with. For a general-purpose use case, the small, default packages are always a good start.

They typically include the following components: Binary weights for the part-of-speech tagger, dependency parser and named entity recognizer to predict those annotations in context. Lexical entries in the vocabulary, i.e. words and their context-independent attributes like the shape or spelling. Data files like lemmatization rules and lookup tables. Word vectors, i.e. multi-dimensional meaning representations of words that let you determine how similar they are to each other. Configuration options, like the language and processing pipeline settings and model implementations to use, to put spaCy in the correct state when you load the pipeline.The figure 4.4 shows processing pipeline of natural language processing. Here Text is converted into doc by tokenising the sentences in the text into words and then into vectors of same length.

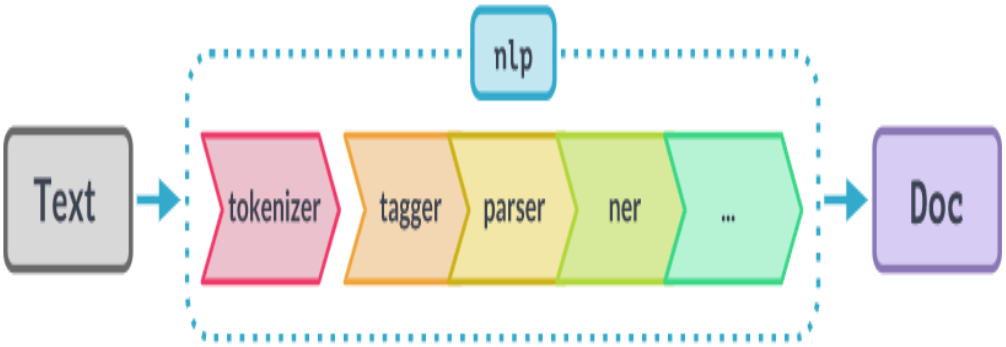

Figure 4.4: Processing pipeline

4.2.4 Python

Python is an interpreted, high-level, general-purpose programming language.Python's design philosophy emphasizes code readability with its notable use of significant white space. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.In this Project mostly uses Python.

Python offers concise and readable code. While complex algorithms and versatile work- flows stand behind machine learning and AI, Python's simplicity allows developers to write reliable systems.Python code is understandable by humans, which makes it easier to build models for Neural Network.

4.2.5 Opencv

Opencv (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.This Project uses Opencv To pre-process the Image data.

4.2.6 Tensorflow

TensorFlow is a free and open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. Tensorflow is a symbolic math library based on dataflow and differentiable programming.

TensorFlow allows developers to create dataflow graphs—structures that describe how data moves through a graph, or a series of processing nodes. Each node in the graph represents a mathematical operation, and each connection or edge between nodes is a multidimensional data array, or tensor.

TensorFlow provides all of this for the programmer by way of the Python language. Python is easy to learn and work with, and provides convenient ways to express how high-level abstractions can be coupled together. Nodes and tensors in TensorFlow are Python objects, and TensorFlow applications are themselves Python applications.

4.2.7 Keras

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.

Keras is the high-level API of TensorFlow 2: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity.

Keras empowers engineers and researchers to take full advantage of the scalability and cross-platform capabilities of TensorFlow 2: you can run Keras on TPU or on large clusters of GPUs, and you can export your Keras models to run in the browser or on a mobile device.

4.2.8 FastAPI

FastAPI is a modern, fast (high-performance), web framework for building APIs with Python 3.6+ based on standard Python type hints. The key features are:

- Fast: Very high performance, on par with NodeJS and Go (thanks to Starlette and Pydantic). One of the fastest Python frameworks available.
- Fast to code: Increase the speed to develop features by about 200
- Fewer bugs: Reduce about 40
- Intuitive: Great editor support. Completion everywhere. Less time debugging.
- Easy: Designed to be easy to use and learn. Less time reading docs.
- Robust: Get production-ready code. With automatic interactive documentation.
- Standards-based: Based on (and fully compatible with) the open standards for APIs: OpenAPI (previously known as Swagger) and JSON Schema.

Chapter 5

Implementation and Testing

This section explains the implementation details of three deeplearning models developed in this project.Here testing,accuracy and loss of each model is explained in detail.

5.1 Task Definition

An instance of VQA consists of an image and a related question given in the plain text. The task for the machine learning model is to determine the correct answer, which is, in current data sets, consists of a few words or a short phrase. Two practical variants of VQA are usually considered, an open-ended and a multiple-choice setting [1], [8]. In the second variant, a set of candidate answers are proposed. It makes the evaluation of a generated answer easier than in the open-ended setting. The comparison between the machine's output and a ground truth that are human-provided answers faces issues with paraphrasing and synonyms.

In comparison to tasks of computer vision such as object recognition or image segmentation, instances of VQA cover a wide range of complexity. Indeed, the question itself can take a random form, and so can the set of operations required to answer it. In this sense, VQA more closely reflects the challenges of general image understanding. VQA is also related to the task of textual question-answering in which the answer is to be found in a textual narrative or large knowledge base. Textual QA has been studied for a long time in the natural language processing (NLP) community, and VQA is basically its extension to visual input. The additional challenge of visual input is significant because images are higher dimensional than text. Images capture the

Modern computer vision is based on statistical learning, and recent works combining vision and language, including image captioning and VQA are similarly evolved white;from machine learning techniques. Finally, both language and vision are inherently compositional in their structure. This constitutes both a challenge and an

opportunity when considering the generalization capabilities of learned models.The Figure 5.1 shows the sample images from major public dataset.

Figure 5.1: The task of VQA is a significant step toward general AI and a departure from low- and mid-level tasks in classical computer vision. It requires relating visual concepts with elements of language, common-sense, and general knowledge. (Photos are examples from a major public data set [1])

There exists a relation between VQA and the task of automatic image captioning [3], [5], [6], that is, generating a textual description of a given image. It has attracted significant interest in the past few years and can be compared to VQA as they both combine vision and language. The two tasks are complementary as they evaluate different capabilities. Captioning requires descriptive capabilities that involve almost purely visual information. VQA, in comparison, often requires reasoning with common sense and with other information not present in the given image. In this respect, VQA constitutes an AI-complete task [1] since it requires multimodal knowledge beyond specific domains. Hence reinforces the motivation for research on VQA, as it provides a proxy to evaluate progress toward general AI, with systems capable of advanced reasoning combined with deep image and language understanding.

5.2 Dataset

In this project there are three visual questions answering models. These are categorized based on the type of images and complexity of the model. First, one being the simple VQA where the dataset contains images of 2-dimensional objects such as squares, circles, and others. The questions are related to color, shape, and the presence/absence of the shape in the image. The second one being the real-time images. The questions in this dataset here are open-ended. The third model is for a specific application which is FloodNet:VQA for Post Flood Scene Understanding. This section briefs about the dataset used in each of the models.

5.2.1 Simple visual question answering dataset(easy vqa)

The easy-VQA dataset[9] is a beginner-friendly way to get started a "Hello World" for VQA. It contains 5k simple, geometric images and 48k questions with only 13 possible answers. Extremely accurate models can be trained on this dataset relatively quickly, even with unimpressive hardware.The questions are also much simpler:The Figure 5.2 shows some of the sample images from the easy-vqa dataset.

- What color is the triangle?
- Does the image contain a circle?
- Is there a green shape in the image?

These questions have 13 possible answers

- Yes/No: Yes, No
- Shapes: Circle, Rectangle, Triangle
- Colors: Red, Green, Blue, Black, Gray, Teal, Brown, Yellow

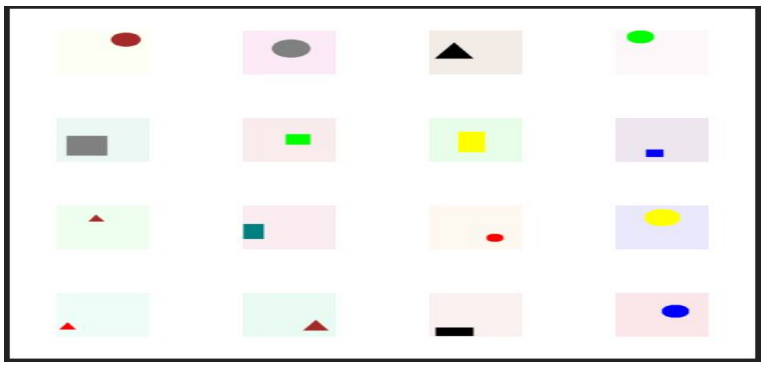

Figure 5.2: 16 example images from easy-VQA[9].

5.2.2 Real time VQA Dataset-VQA2.0

VQA 2.0 [10] is a new dataset containing open-ended questions about images. These questions require an understanding of vision, language and commonsense knowledge to answer. In VQA 2.0, for each (I, Q, A), where the three tuple are the image, question, and answer, another image I 0 were found,where question Q makes sense for the image I 0 , however the answer for the question will be something different, A 0 . Understandably, for some of the images finding this counter example image-question pair was not possible.However, by adding these new images to the dataset, the imapct of language priors on the models were mitigated to a great degree, as compared to first version of VQA dataset.The figure 5.3 shows sample images from dataset VQA 2.0 taken from vqa.org.

Figure 5.3: Sample images from VQA Dataset[10].

This dataset contains:

- 265,016 images
- At least 3 questions (5.4 questions on average) per image
- 10 ground truth answers per question
- 3 plausible (but likely incorrect) answers per question

However In this only 50000 images and 150000 questions are randomly selected. This is done because of the constrained hardware. For each image there are 3 questions which are related to that image.

5.2.3 Floodnet : VQA for Post Flood Scene Understanding

FloodNet provides high-resolution UAS imageries with detailed semantic annotation regarding the damages. To advance the damage assessment process for post-disaster scenarios, this presents a unique challenge considering classification, semantic segmentation, visual question answering highlighting the UAS imagery-based FloodNet dataset.The data is collected with a small UAS platform, DJI Mavic Pro quadcopters, after Hurricane Harvey. The figure 5.4 shows the sample images taken from Floodnet Dataset.

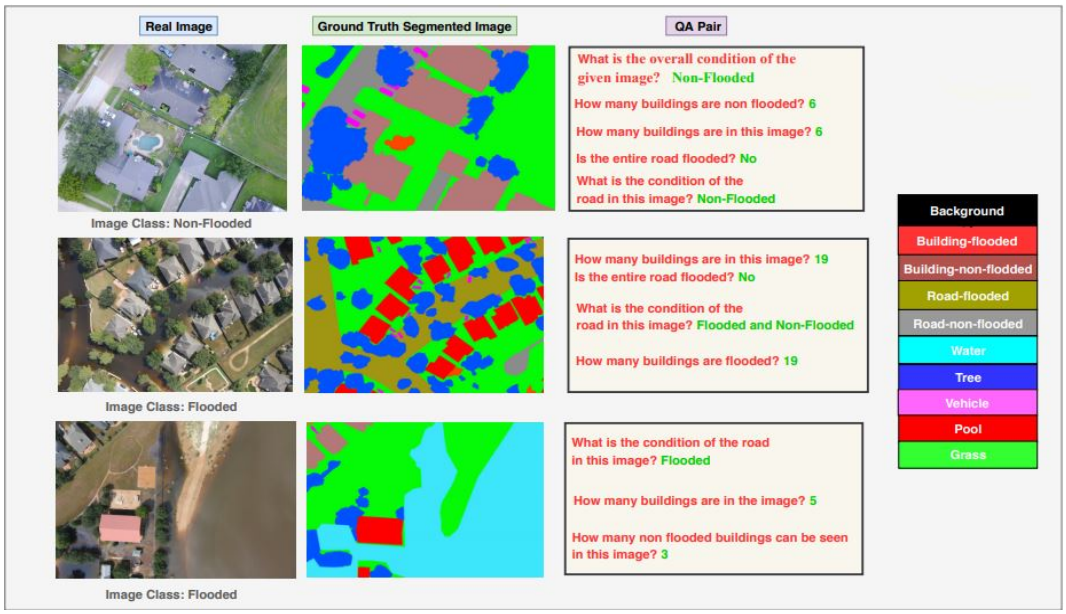

Figure 5.4: Sample images from Floodnet dataset[11].

The whole dataset has 1450 images and there are a total 4511 image-question pairs divided into training (60%), validation (20%), and test (20%) sets.These questions will be divided into the following categories:

- Simple Counting: Questions will be designed to count the number of objects regardless of their attribute. For example: “how many buildings are there in the image?”.
- Complex Counting: Questions will be asked to count the number of objects belonging to a specific attribute. For example: “how many flooded buildings are there in the image?”.

- Condition Recognition: In this category, all the questions are mainly designed to ask questions regarding the condition of the object and the neighborhood. For example: “What is the condition of the road in the given image?”.
- Yes/No type of question: For this type of question, the answer will be either a ‘Yes’ or a ‘No’. For example: “Is there any flooded road?”.

5.3 Implementation and testing of the model

In this project, there are three models for visual question answering. The following sections contain an explanation about the implementation and testing of each of the models.

5.3.1 Implementation of Simple visual question answering

The standard approach to performing VQA looks something like this,Process the image,Process the question,Combine features from previous two,Assign probabilities to each possible answer.Here there are fixed answer set where exactly one of the possible answers is guaranteed to be correct. It’s just have to answer what is effectively a multiple-choice question, but this model will only allow the 13 possible answers included in easy-VQA[9].Processing of the image and question generally use methods from Computer Vision and Natural Language Processing, respectively, to turn raw image / text inputs into processed data vectors.

These two output representations can then be used analyzed together to ultimately pick the most likely answer. Using TensorFlow, which comes packaged with Keras as its high-level API, and using Pillow for image processing. With the help of the easy-vqa Python library, which makes accessing the easy-VQA dataset a breeze.The figure 5.5 shows the Convolutional Neural Network that is applied for this model.

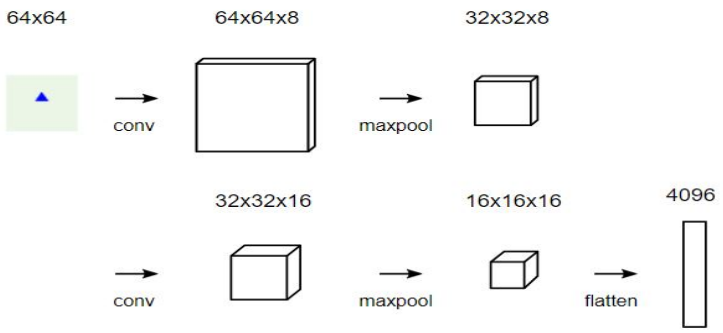

Figure 5.5: CNN Easy VQA

First to capture image features a Convolutional Neural Network (CNN) to extract information from the image input. To do this,Keras, a powerful deep learning library for Python is used.This image dataset is not very complex, so it can be tackled it with a relatively simple.Following are the steps,Start with a 64x64 image from the dataset.Pass through a conv layer with eight 3x3 filters using “same” padding, resulting in a 64x64x8 volume.Use a standard max pooling layer to cut the volume to 32x32x8.Pass through another conv layer, this time with 16 filters, resulting in a 32x32x16 volume.Use max pooling again, cutting to 16x16x16.Flatten the volume, which results in a layer with 16**3 = 4096 nodes.

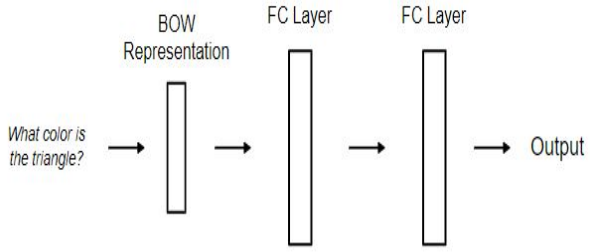

Figure 5.6: Bag of words Easy VQA

Next is to extract question features. Most VQA models would use some kind of Recurrent Neural Network (RNN) to process the question input, but that’s a little overkill for this use case. The questions in the easy-VQA dataset are short, simple, and come from a fixed set of question templates, so they’re much more approachable compared to those you might see in the real world.

Here Bag of Words (BOW) representation is used to turn each question into a vector and use that vector as input to a feed forward neural network.A BOW representation turns any text string into a fixed-length vector by counting how many times each word appears in the text.The figure 5.6 shows the representation of words in the easy vqa model.

As discussed before, our question dataset is relatively simple, so ther is no need of anything too fancy for our question model. Just pass the BOW vector representation into 2 fully-connected (FC) neural network layers.This will use a very simple method to merge image and question vectors: element-wise multiplication.Finally, it’s time for VQA system to produce an answer. Recall that this project is working with fixed answer set: We know every conceivable answer, and only one is guaranteed to be correct. For this step, with the help of Softmax to turn our output values into probabilities allowing us to quantify how sure each possible answer will be.The figure 5.7 shows the Summary of Easy VQA model , it shows the shape of

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 64, 64, 3)]	0	
conv2d (Conv2D)	(None, 64, 64, 8)	224	input_1[0][0]
max_pooling2d (MaxPooling2D)	(None, 32, 32, 8)	0	conv2d[0][0]
conv2d_1 (Conv2D)	(None, 32, 32, 16)	1168	max_pooling2d[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 16)	0	conv2d_1[0][0]
conv2d_2 (Conv2D)	(None, 16, 16, 32)	4640	max_pooling2d_1[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 8, 8, 32)	0	conv2d_2[0][0]
input_2 (InputLayer)	[(None, 27)]	0	
flatten (Flatten)	(None, 2048)	0	max_pooling2d_2[0][0]
dense_1 (Dense)	(None, 32)	896	input_2[0][0]

Figure 5.7: Summary of Easy VQA model

the output in each layer.The figure 5.8 and figure 5.9 shows loss and accuracy of the easy VQA model.After training, validation accuracy reached 83.92%. Training progress is easily visible (losses decreasing, accuracies increasing) the model isn't overfitting too badly (training/validation losses and accuracies are close enough).

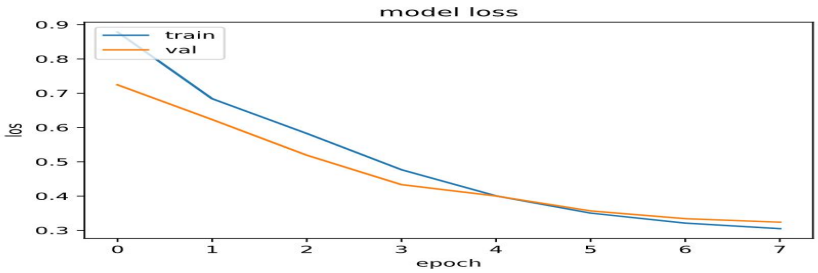

Figure 5.8: Simple VQA model loss

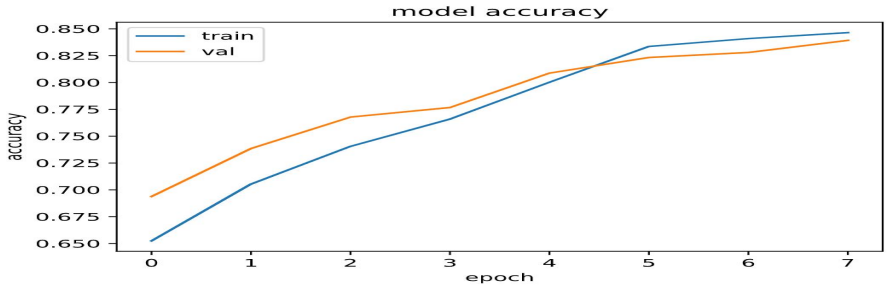

Figure 5.9: Simple VQA model accuracy

5.3.2 Implementation of Real time VQA

For our methods, a two channel vision (image) + language (question) model was developed that culminates with a softmax over K possible outputs. Selecting the top K = 1000 most frequent answers as possible outputs. This set of answers covers most of the train+val answers. Describing the different components of our model below:

Image Channel: This channel provides an embedding for the image. Experimented with following embeddings I: The activations from the last hidden layer of VGGNet[10] are used as 4096-dim image embedding.

Question Channel: This channel provides an embedding for the question. The following embeddings were used: Deeper LSTM Q: An LSTM with two hidden layers is used to obtain 2048-dim embedding for the question. The embedding obtained from the LSTM is a concatenation of last cell state and last hidden state representations (each being 512-dim) from each of the two hidden layers of the LSTM. Hence 2 (hidden layers) x 2 (cell state and hidden state) x 512 (dimensionality of each of the cell states, as well as hidden states). This is followed by a fully-connected layer + tanh non-linearity to transform 2048-dim embedding to 1024-dim. The question words are encoded in the same way as in LSTM Q.The Figure 5.10 shows model architecture that is implemented in this Model . The architecture is similar with minute changes in the number of units in each layer.

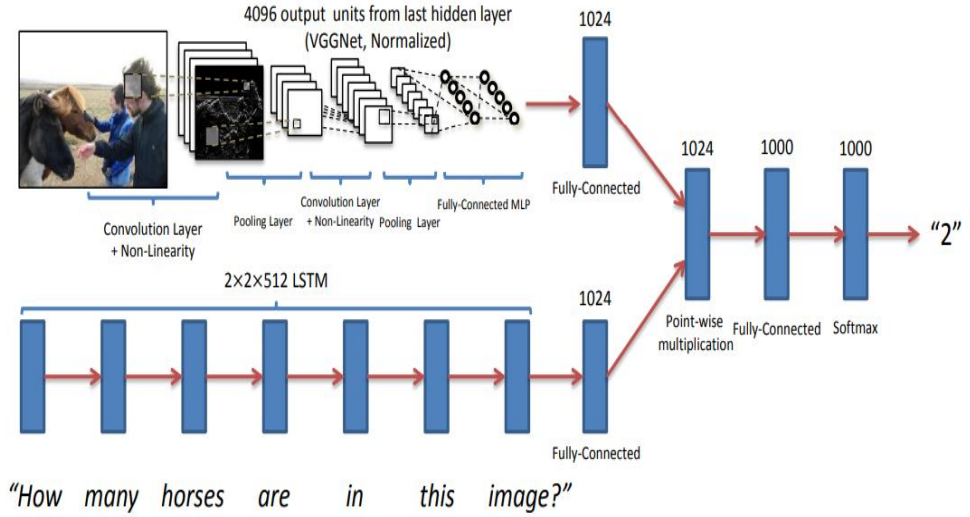

Figure 5.10: Similar Model Architecture of the real time vqa

Multi-Layer Perceptron (MLP): The image and question embeddings are combined to obtain a single embedding.) For ddeeper LSTM Q + I method, the image embedding is first transformed to 1024-dim by a fully-connected layer + tanh nonlinearity to match the LSTM embedding of the question. The transformed image and LSTM embeddings (being in a common space) are then fused via element-wise multiplication.

This combined image + question embedding is then passed to an MLP – a fully connected neural network classifier with 2 hidden layers and 1000 hidden units (dropout 0.5) in each layer with tanh non-linearity, followed by a softmax layer to obtain a distribution over K answers. The entire model is learned end-to-end with a cross-entropy loss. VGGNet parameters are frozen to those learned for ImageNet classification and not fine-tuned in the image channel.

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
img_input (InputLayer)	[(None, 1, 4096)]	0	
dense (Dense)	(None, 1, 2048)	8390656	img_input[0][0]
ques_input (InputLayer)	[(None, 25, 300)]	0	
dense_1 (Dense)	(None, 1, 1024)	2098176	dense[0][0]
ques_reshape (Reshape)	(None, 25, 300)	0	ques_input[0][0]
vision_model (Reshape)	(None, 1024)	0	dense_1[0][0]
ques_model (LSTM)	(None, 1024)	5427200	ques_reshape[0][0]
multiply (Multiply)	(None, 1024)	0	vision_model[0][0] ques_model[0][0]
dense_2 (Dense)	(None, 1024)	1049600	multiply[0][0]
dropout (Dropout)	(None, 1024)	0	dense_2[0][0]
dense_3 (Dense)	(None, 1024)	1049600	dropout[0][0]
dropout_1 (Dropout)	(None, 1024)	0	dense_3[0][0]
dense_4 (Dense)	(None, 14088)	14440200	dropout_1[0][0]

Total params: 32,455,432
Trainable params: 32,455,432
Non-trainable params: 0

Figure 5.11: Summary of the Real Time VQA Model

The accuracy of our model selected using VQA test-dev accuracies) on VQA teststandard is 74% It is clearlu visible that our model is able to significantly outperform both the vision-alone and language-alone baselines.The Figure 5.12 and Figure 5.13 shows the loss and accuracy curve of the real time vqa model.

Figure 5.12: Realtime VQA model loss

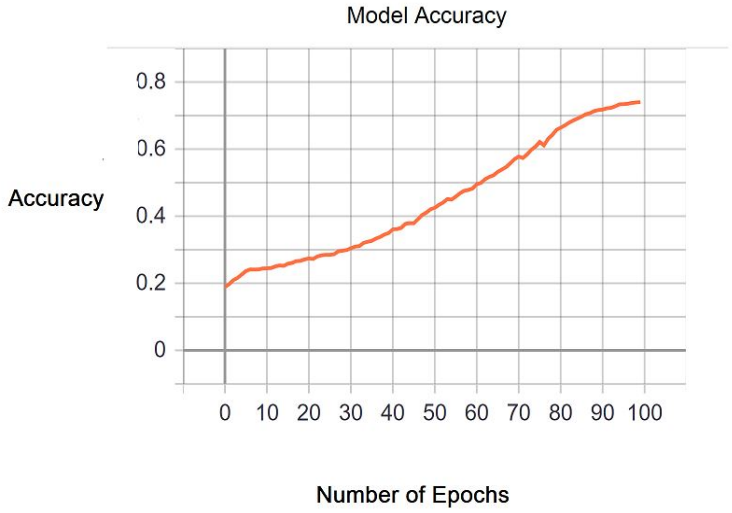

Figure 5.13: Realtime VQA model accuracy

5.3.3 Implementation of Floodnet VQA

In this model, using FloodNet[11] dataset to build and test VQA algorithms that can be implemented during natural emergencies. To the best of our knowledge, this is the first VQA dataset focused on UAV imagery for disaster damage assessment. To evaluate the performances of existing VQA algorithms, baseline models on this dataset was implemented followed by the implementation of the same model with minor modifications in the architecture.This section just presents the summary and the results.The Figure 5.14 shows the summary of the Floodnet visual question answering implemented in this model.

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
img_input (InputLayer)	[(None, 1, 4096)]	0	
dense_2 (Dense)	(None, 1, 2048)	8390656	img_input[0][0]
ques_input (InputLayer)	[(None, 15, 300)]	0	
dense_3 (Dense)	(None, 1, 1024)	2098176	dense_2[0][0]
ques_reshape (Reshape)	(None, 15, 300)	0	ques_input[0][0]
vision_model (Reshape)	(None, 1024)	0	dense_3[0][0]
ques_model (LSTM)	(None, 1024)	5427200	ques_reshape[0][0]
multiply (Multiply)	(None, 1024)	0	vision_model[0][0] ques_model[0][0]
dense_4 (Dense)	(None, 1024)	1049600	multiply[0][0]
dropout (Dropout)	(None, 1024)	0	dense_4[0][0]
dense_5 (Dense)	(None, 1024)	1049600	dropout[0][0]
dropout_1 (Dropout)	(None, 1024)	0	dense_5[0][0]
dense_6 (Dense)	(None, 41)	42025	dropout_1[0][0]

Total params: 18,057,257
Trainable params: 18,057,257
Non-trainable params: 0

Figure 5.14: Summary of the Model Floodnet vqa

This model has a training accuracy of 93.05% and validation accuracy of 80.33%. This model is comparatively simpler to train and provides better accuracy , because of the smaller dataset compared to real time visual question answering.

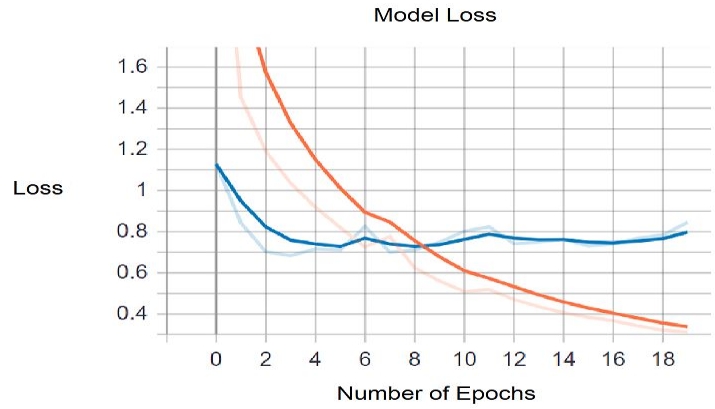

Figure 5.15: Floodnet VQA model loss

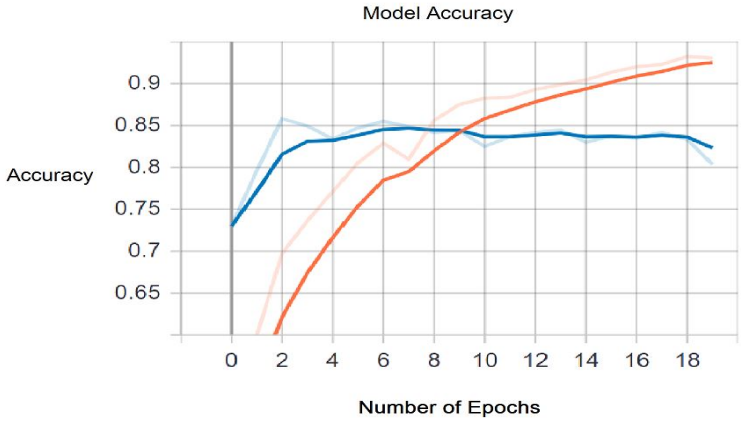

Figure 5.16: Floodnet VQA model accuracy

The Figure 5.15 and Figure 5.16 shows the loss and accuracy of the floodnet VQA model.Here it can be observed that the loss decreases and accuracy increses as the number of epochs increases. This model is trained for twenty epochs.

5.3.4 Demo

The models are deployed on the web . Here a React web application was developed for frontend website and utilizing Fast api - A python based web framework for developing the rest api. The frontend will talk with rest api.In React front end one can select the local images or upload their own image. The question can be given in the form of text or use microphone. Speech to text library was used which converts speech to text.The backend rest api is containerized using Docker. This Docker container is deployed on Google Cloud Run . Google Cloud Run is a managed

compute platform that enables you to run containers that are invocable via requests or events. Cloud Run is serverless: it abstracts away all infrastructure management, so you can focus on what matters most:building great applications.The figure 5.17 shows the Easy visual question answering model that is deployed on web and Figure 5.18 shows the Real time question answering model delployed on the Internet.

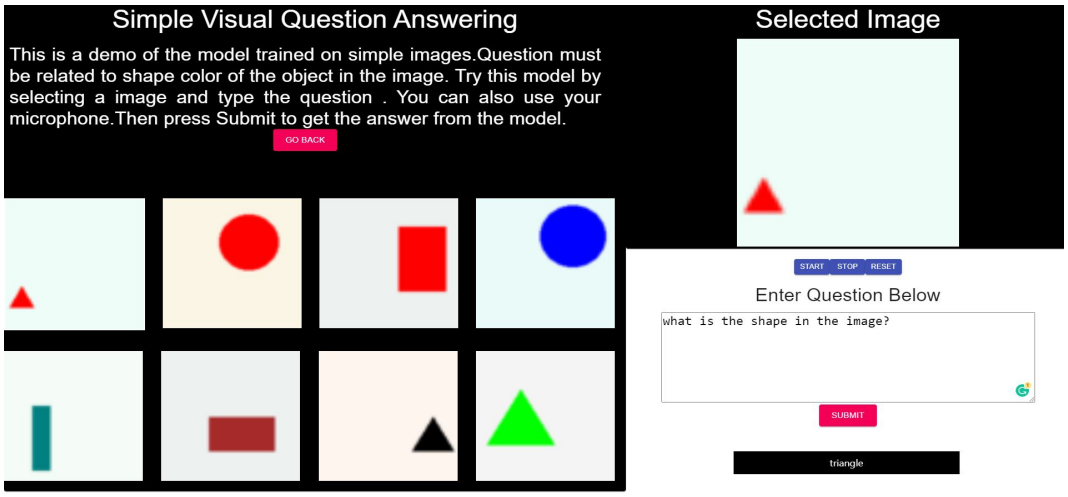

Figure 5.17: Simple Visual Question Answering model deployed on web

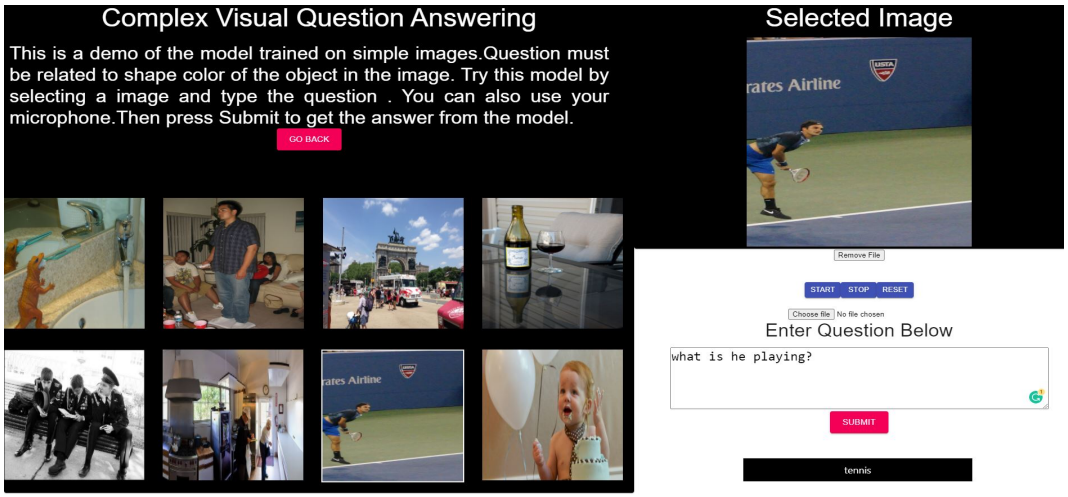

Figure 5.18: Real time Visual Question Answering model deployed on web

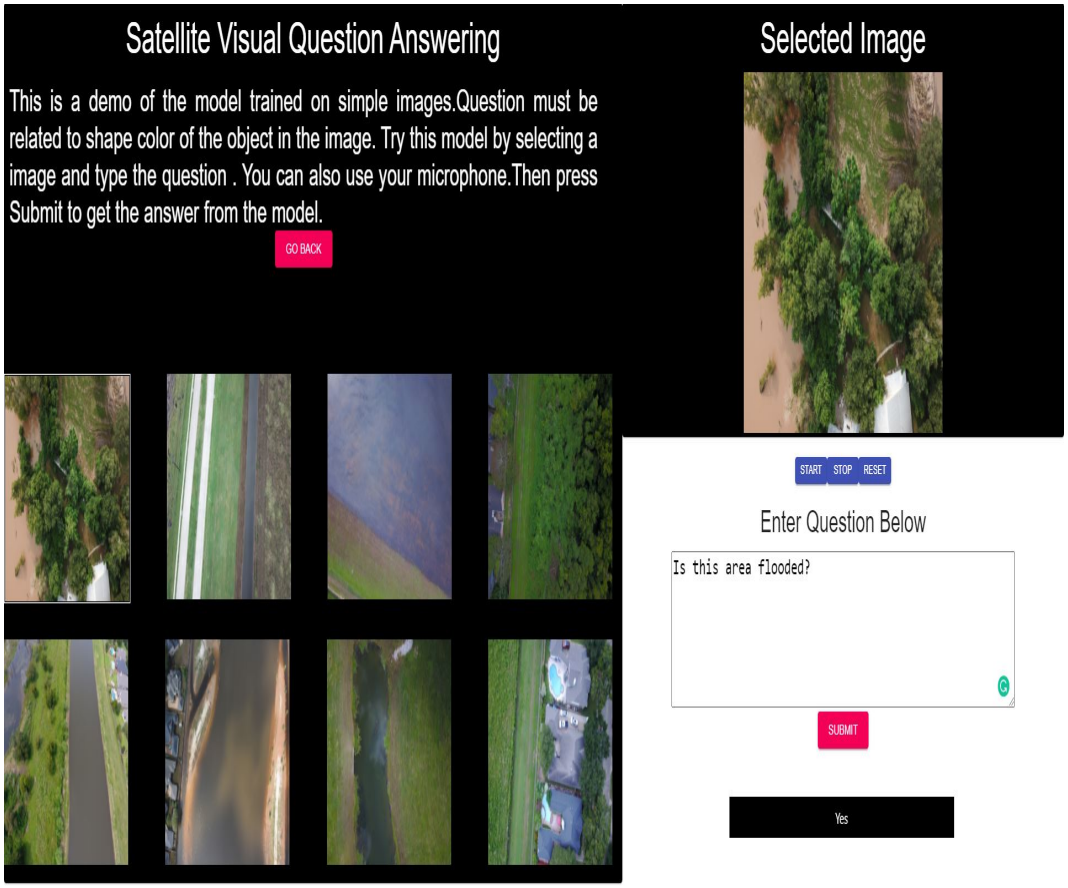

Figure 5.19: Floodnet : VQA for Post Flood Scene Understanding

The backend fastpi deployed on google cloud run is on the site "https://sjce-vqa-api-xueksbf2aa-el.a.run.app" and the frontend web application is on "https://visual-ques-ans.herokuapp.com".Anyone can try this. This site is not responsive. Therefore it is recommended to use this site only on a laptop/desktop.The Figure 5.19 shows the Floodnet Visual Question Answering that is on the site mentioned above. In all these model user can select an image and type the question to get results.There is also a provision where the user can input the question via microphone.

Chapter 6

Conclusion

In conclusion, The task of Visual Question Answering is introduced(VQA). Given an image and an open-ended, natural anguage question about the image, the task is to provide an accurate natural language answer.We demonstrate the wide variety of questions and answers in our dataset, as well as the diverse set of AI capabilities in computer vision, natural language processing,and commonsense reasoning required to answer these questions accurately. VQA is believed to have a distinctive advantage of pushing the frontiers on "AI-complete" problems, while being amenable to automatic evaluation. Given the recent progress in the community, this is the time is ripe to take on such an endeavor.The Table 6.1 shows the Deep learning models implemented in this project and their respective accuracy.

Model	Accuracy in %
Simple VQA	84
Real time VQA	74
Floodnet	80

Table 6.1: Model and their respective accuracy

6.1 Advantages

In a general, a VQA system can be defined as an algorithm that takes as input an image and a natural language question about the image and generates a natural language answer as the output. This is by nature a multi-discipline research problem.NLP is needed for at least two reasons: to understand the question and to generate the answer. These are common problems in text-based Q and A, a well studied problem in NLP. Given the following sentence: How many bridges are there in Paris? a NLP Q and A system is typically going to:Classify or type the question: this is a "how many" question, so the response must be a number.Extract the object to count: bridges.Extract the context where the count must be performed: in

this case, the city of Paris.After the question has been analyzed, the system builds some kind of query and relies on a knowledge base to get the answer. This is far from trivial (e.g. there are at least 22 cities called Paris in the United States), but a lot of work has been done since the 1970s.The main difference in VQA is that the search and the reasoning part must be performed over the content of an image.

So, to answer if there are any humans, the system must be able to detect objects. To say if it is raining, it needs to classify a scene. To answer who are the teams, the system needs world knowledge. Finally, to say which player has the ball, commonsense reasoning and, very likely, knowledge reasoning are necessary. Many of these tasks (object recognition, object detection, scene classification, etc.) have been addressed in the field of Computer Vision (CV), with impressive results in the last few years. A good VQA system must be capable of solving a broad spectrum of typical NLP and CV tasks, as well as reasoning about image content. It is clearly a multi-discipline AI research problem, involving CV, NLP and Knowledge Representation and Reasoning (KR).

6.2 Limitations and Future Work

While our computational and time resources were limited as a result of class deadlines and budget,it was still possible to begin an extensive architecture and hyperparameter search.Our future work would look at the synergistic effects of some of these hyperparameters , as well as experiment with how a attention mechanism would impact performance. It's desirable to implement this for videos.Visual question answering is a unique challenge in modern Artificial Intelligence research as it combines learning's from both computer vision and natural language processing .

This paper presents the implementation of visual question answering in the real world.Hoping the insights learned from the completion of this project will help us further progress in this task.

Bibliography

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Visual question answering," in Proc. IEEE Int. Conf. Computer Vision, 2015, pp. 2425–2433.

[2] Gong, Yunchao, Wang, Liwei, Guo, Ruiqi, and Lazebnik, Svetlana. Multi-scale orderless pooling of deep convolutional activation features. In ECCV. 2014.

[3] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, and J. Platt, "From captions to visual concepts and back," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2015, pp. 1473–1482.

[4] Z. Zhao, S. Xiao, Z. Song, C. Lu, J. Xiao and Y. Zhuang, "Open-Ended Video Question Answering via Multi-Modal Conditional Adversarial Networks," in IEEE Transactions on Image Processing, vol. 29, pp. 3859-3870, 2020.

[5] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2014, pp. 3156–3164.

[6] Q. Wu, C. Shen, A. v. d. Hengel, L. Liu, and A. Dick, "What value do explicit high level concepts have in vision to language problems?" in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2016, pp. 203–212.

[7] Kim JH, On KW, Lim W, Kim J, Ha JW, Zhang BT (2016b) Hadamard product for low-rank bilinear pooling. arXiv preprint arXiv :1610.04325 Kiros R, Zhu Y, Salakhutdinov RR, Zemel R, Urtasun R, Torralba A, Fidler S (2015) Skip-thought vectors. In: Advances in neural information processing systems, pp. 3294–3302

[8] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei, "Visual7W: Grounded question answering in images," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2016, pp. 4995–5004.

[9] Victor Zhou,Phillip Wang,"Easy Visual Question Answering A gentle introduction to Visual Question Answering (VQA) using neural networks". `https://victorzhou.com/blog/easy-vqa/`

[10] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In CVPR, volume 1, page 9, 2017

[11] Rahnemoonfar, Maryam and Chowdhury, Tashnim and Sarkar, Argho and Varshney, Debvrat and Yari, Masoud and Murphy, Robin.FloodNet: A High Resolution Aerial Imagery Dataset for Post Flood Scene Understanding.arXiv preprint arXiv:2012.02951,2020

[12] [Zhang et al.2016] Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Yin and Yang:

[13] [Malinowski and Fritz2017] Mateusz Malinowski and Mario Fritz. 2017. A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input

[14] [Fukui et al.2016] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In EMNLP. 1, 2