MOTIVATION

- Visual Turing Test - An AI-complete task. The specificity of the questions enable automatic evaluation.
- Helping the visually impaired - Apps that employ humans to answer visual questions sent by visually impaired people.

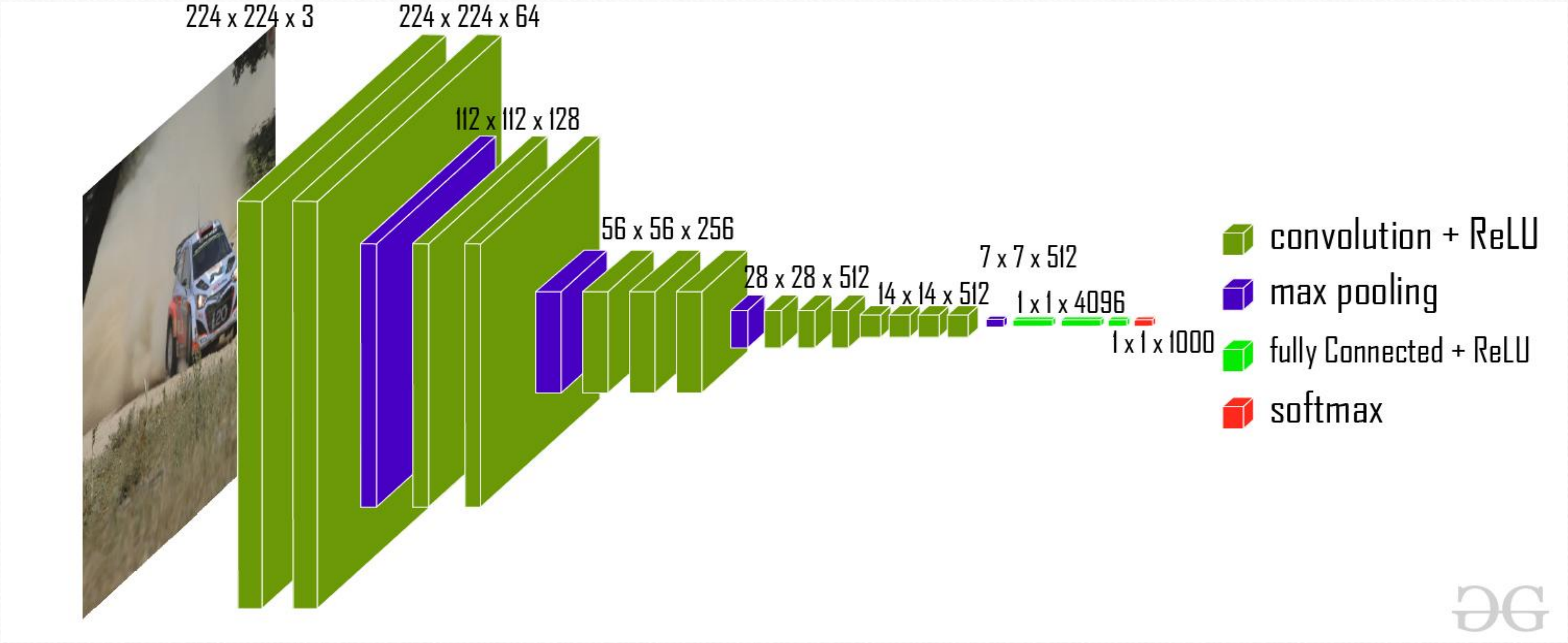
OBJECTIVES

- To Collect data set of images and questions related to image.
- To Preprocess the image data and text data.
- To Build a functional visual questioning model which takes input as image and question.
- To Develop web application which hosts this model.
- To Deploy this web application on the internet.

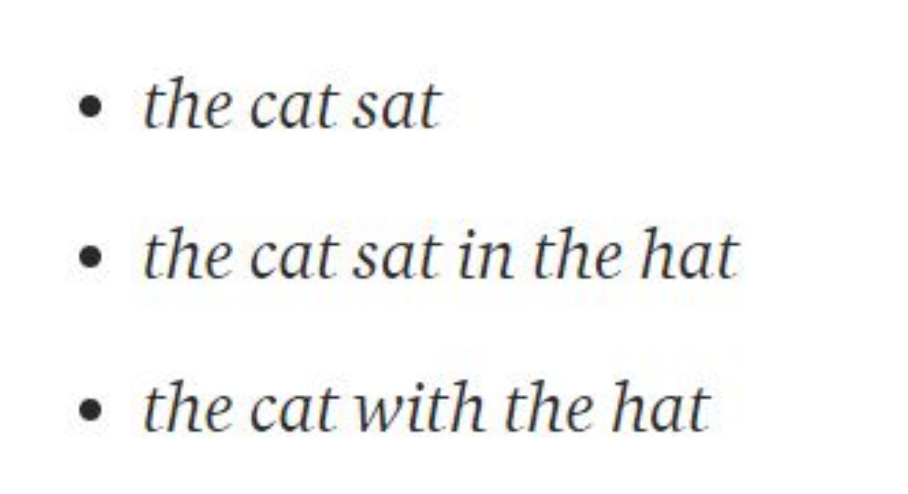
METHODOLOGY

- For this model we use the following design , 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 species both the modules needed to answer the question and the connections between them.
- Figure gives an example of layout designed by this model. The final model combines the result from neural module network with a LSTM question encoder.

VGG Architecture

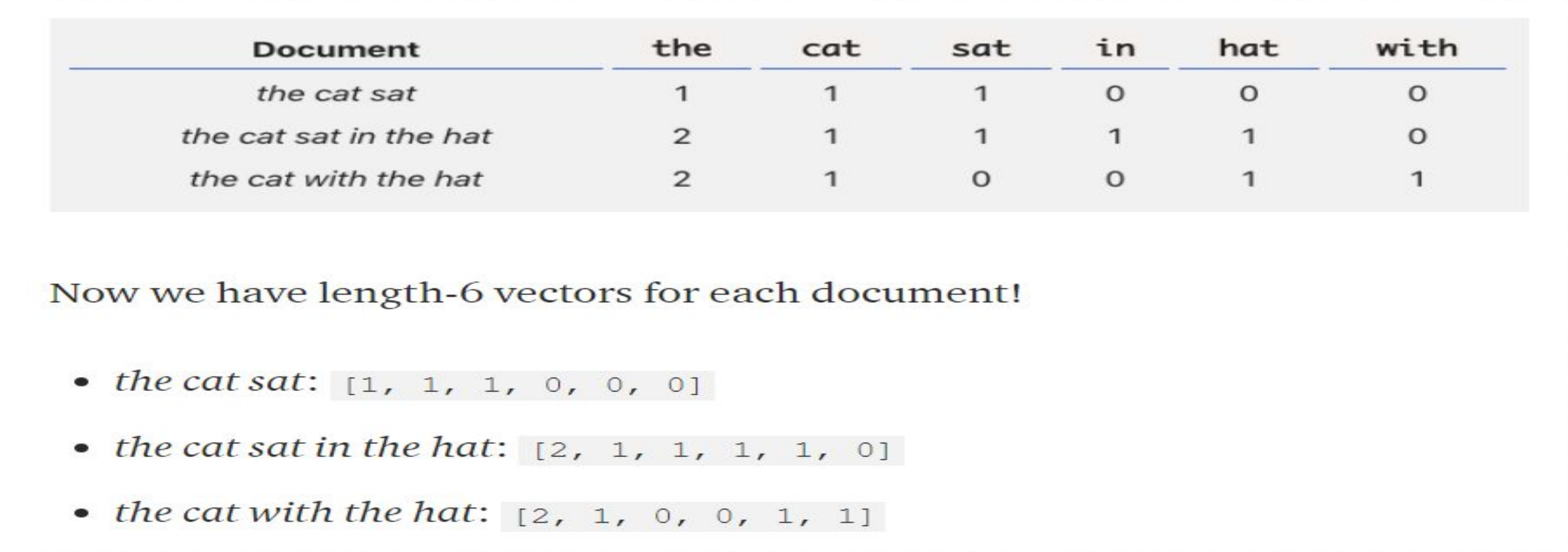


Bag of Words



- *the cat sat*
- *the cat sat in the hat*
- *the cat with the hat*

Bag of Words

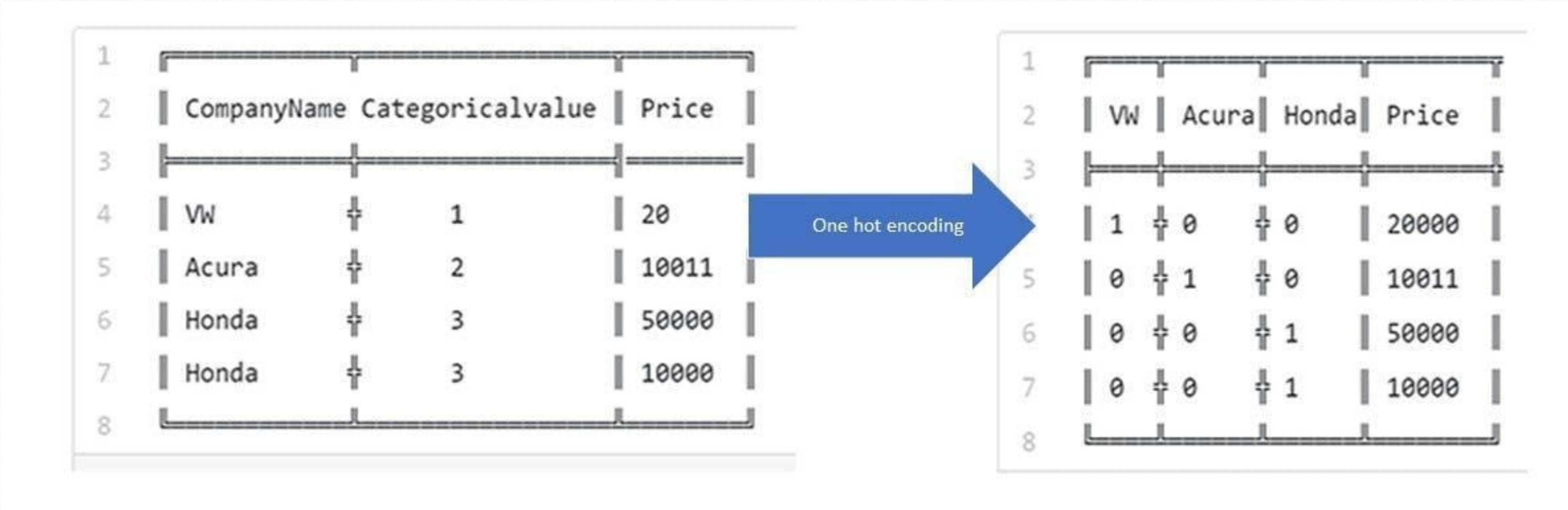


Document	the	cat	sat	in	hat	with
the cat sat	1	1	1	0	0	0
the cat sat in the hat	2	1	1	1	1	0
the cat with the hat	2	1	0	0	1	1

Now we have length-6 vectors for each document!

- *the cat sat*: `[1, 1, 1, 0, 0, 0]`
- *the cat sat in the hat*: `[2, 1, 1, 1, 1, 0]`
- *the cat with the hat*: `[2, 1, 0, 0, 1, 1]`

One Hot Encoding



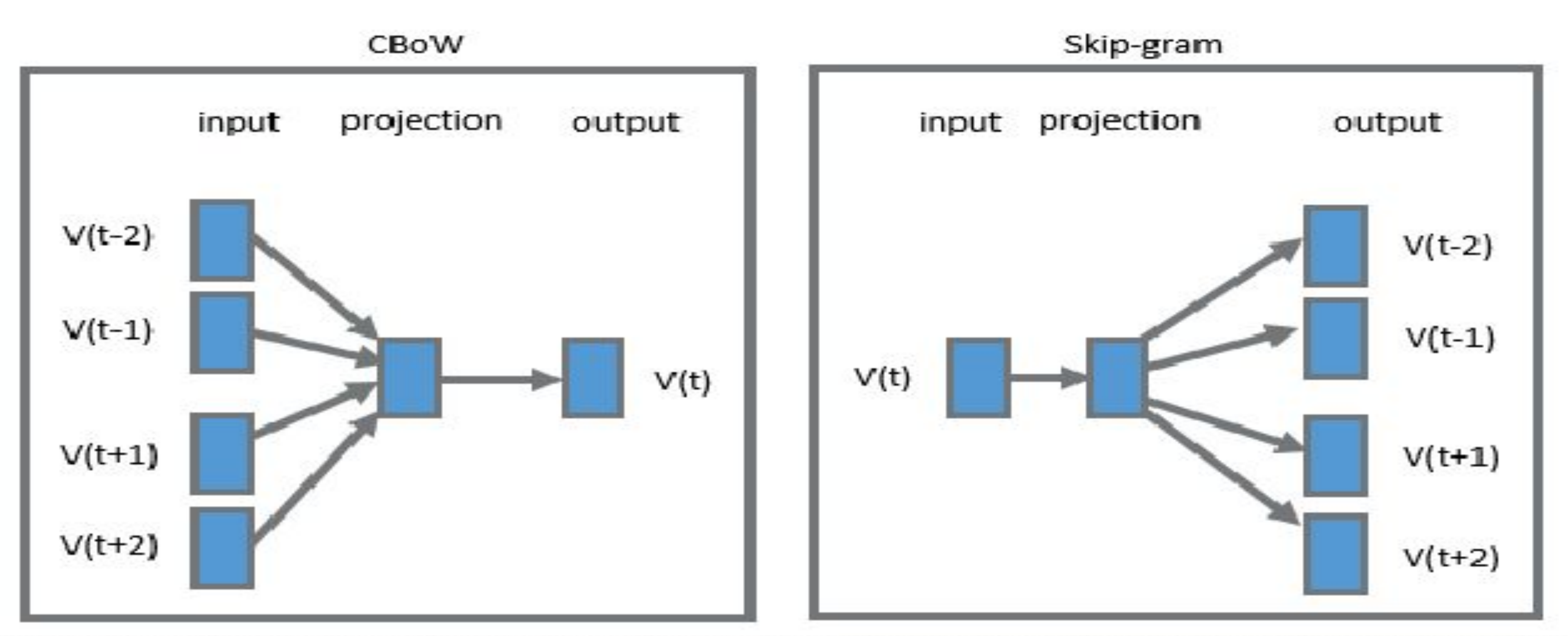
CompanyName	Categoricalvalue	Price
VW	1	20
Acura	2	10011
Honda	3	50000
Honda	3	10000

One hot encoding

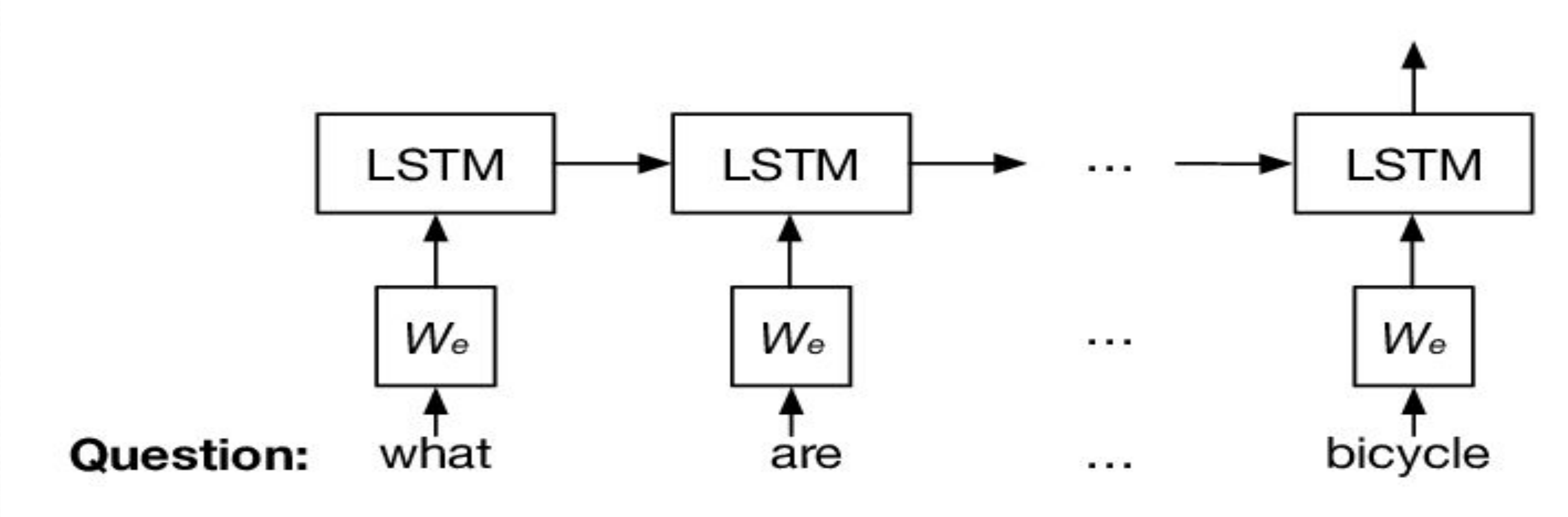
VW	Acura	Honda	Price
1	0	0	20000
0	1	0	10011
0	0	1	50000
0	0	1	10000

Word2Vec Embedding

(Ex: “I am doing good”)



LSTM Question Representation



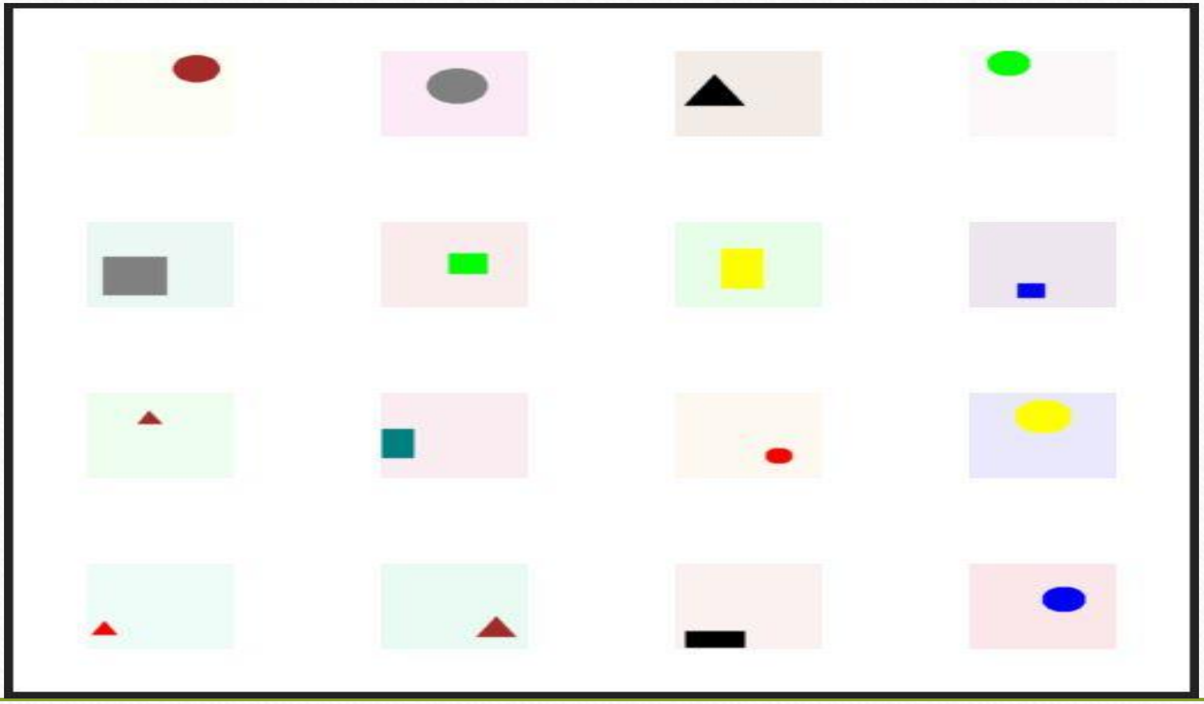
Hardware Software Components

- Graphical Processing Unit
- Tensorflow
- Keras
- Spacy

- FastAPI
- Google Colab
- Docker
- Google Cloud Run

Implementation- Easy VQA

DATASET

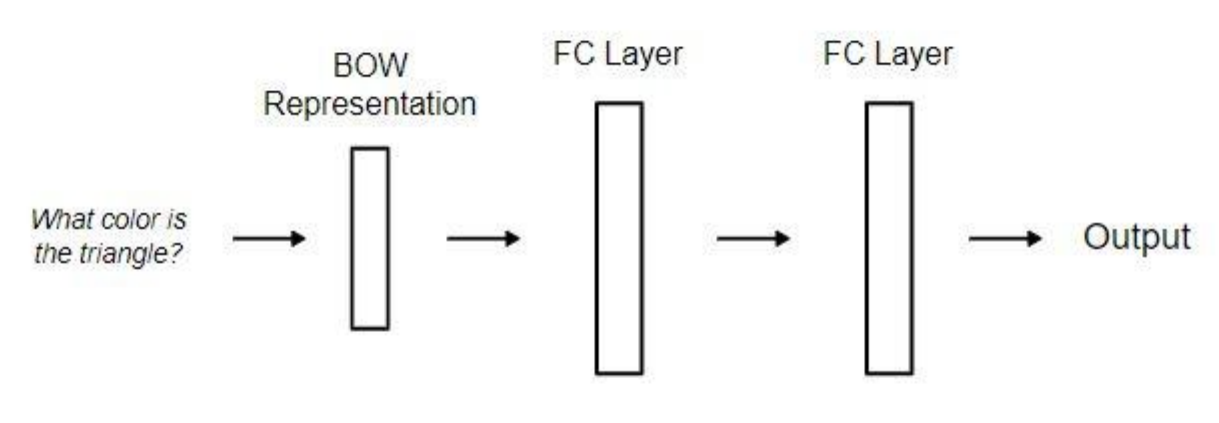


Parameters

- 5000 Images
- 48000 Questions
- 13 Possible Answers

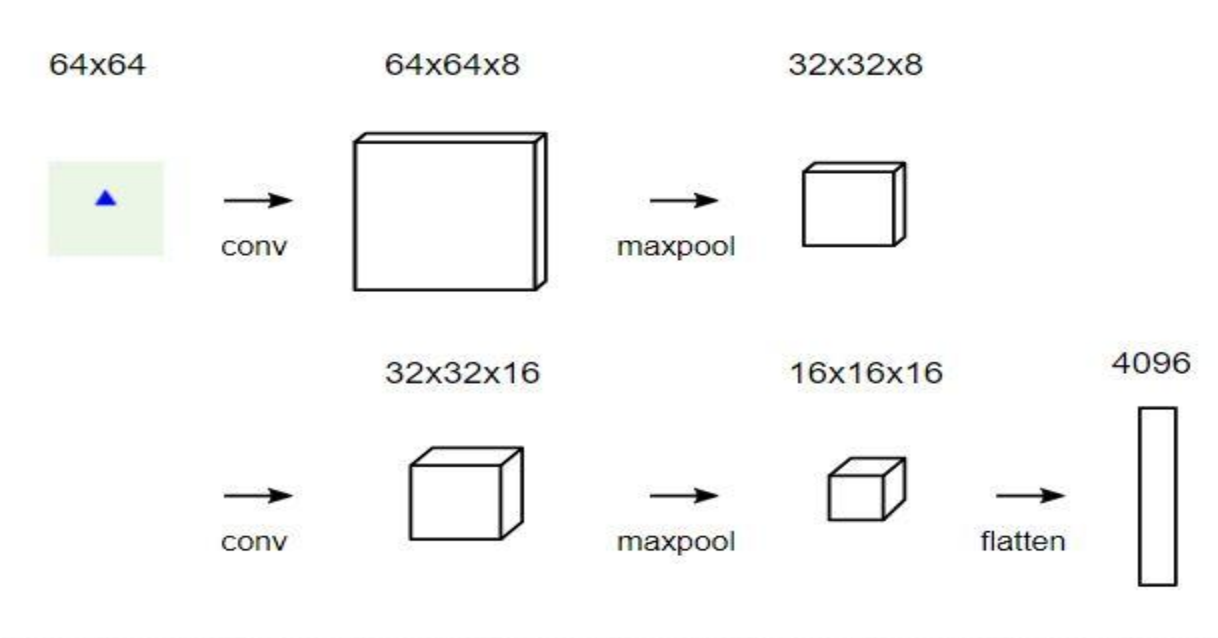
Implementation- Easy VQA

Question Representation



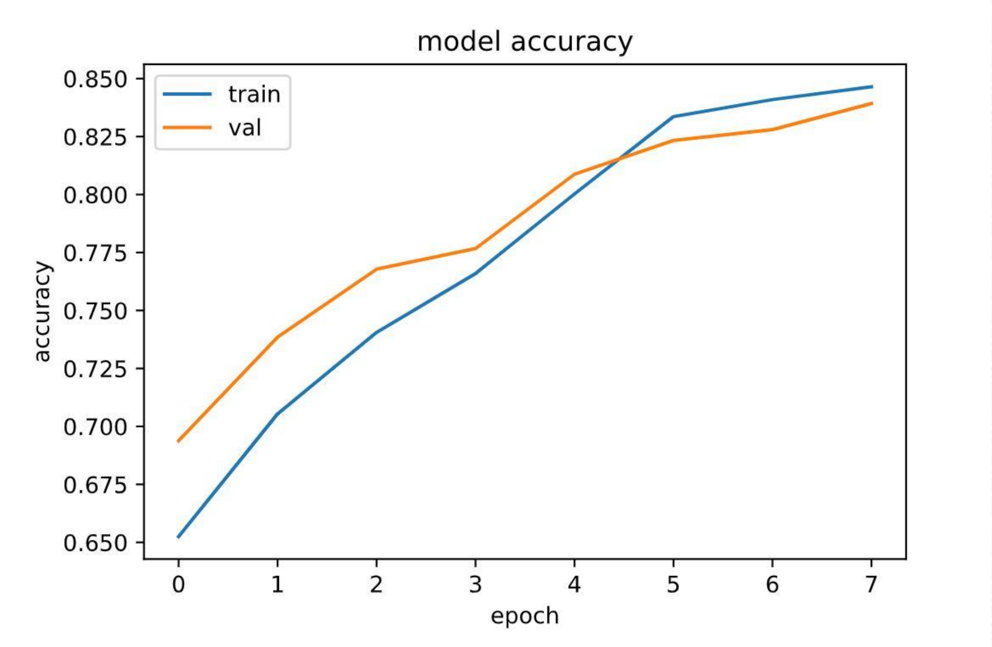
Implementation- Easy VQA

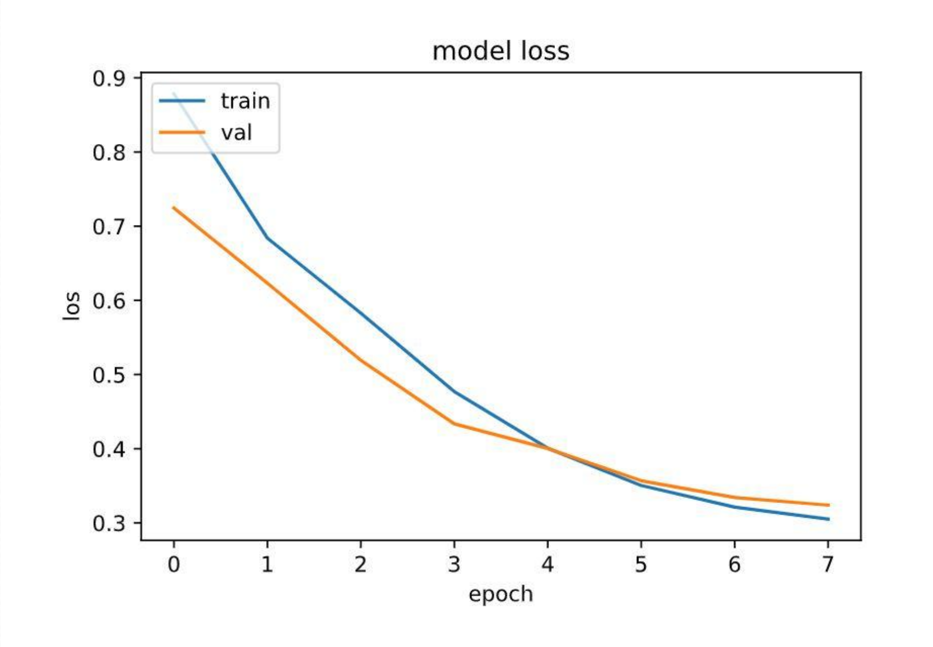
Image Representation



Implementation- Easy VQA

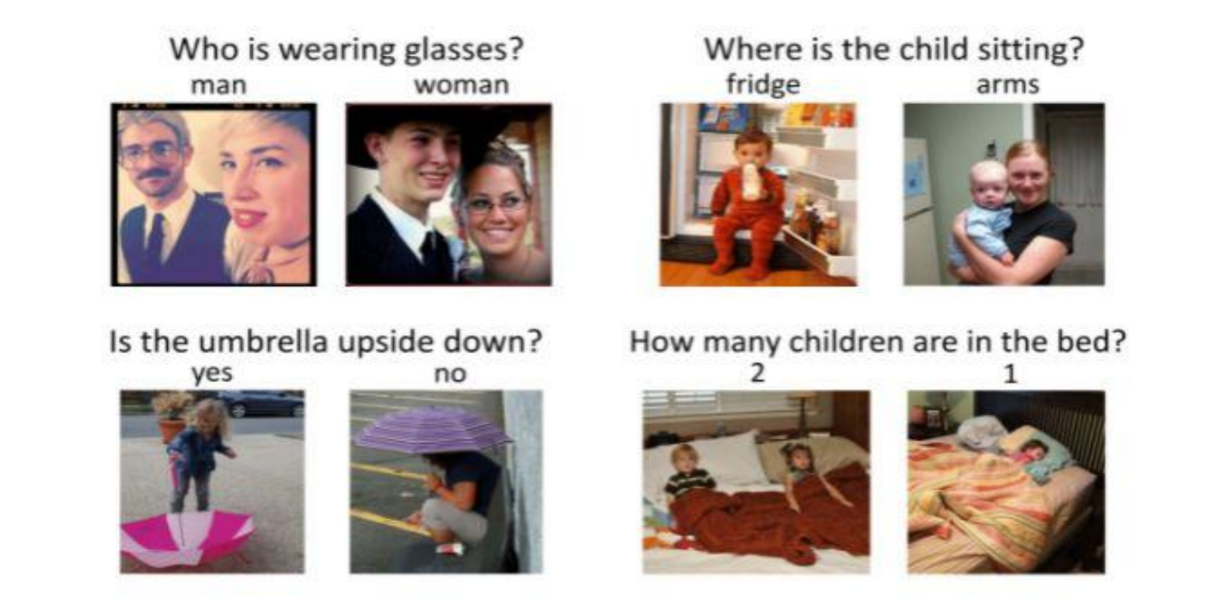
Results





Implementation- Real Time VQA

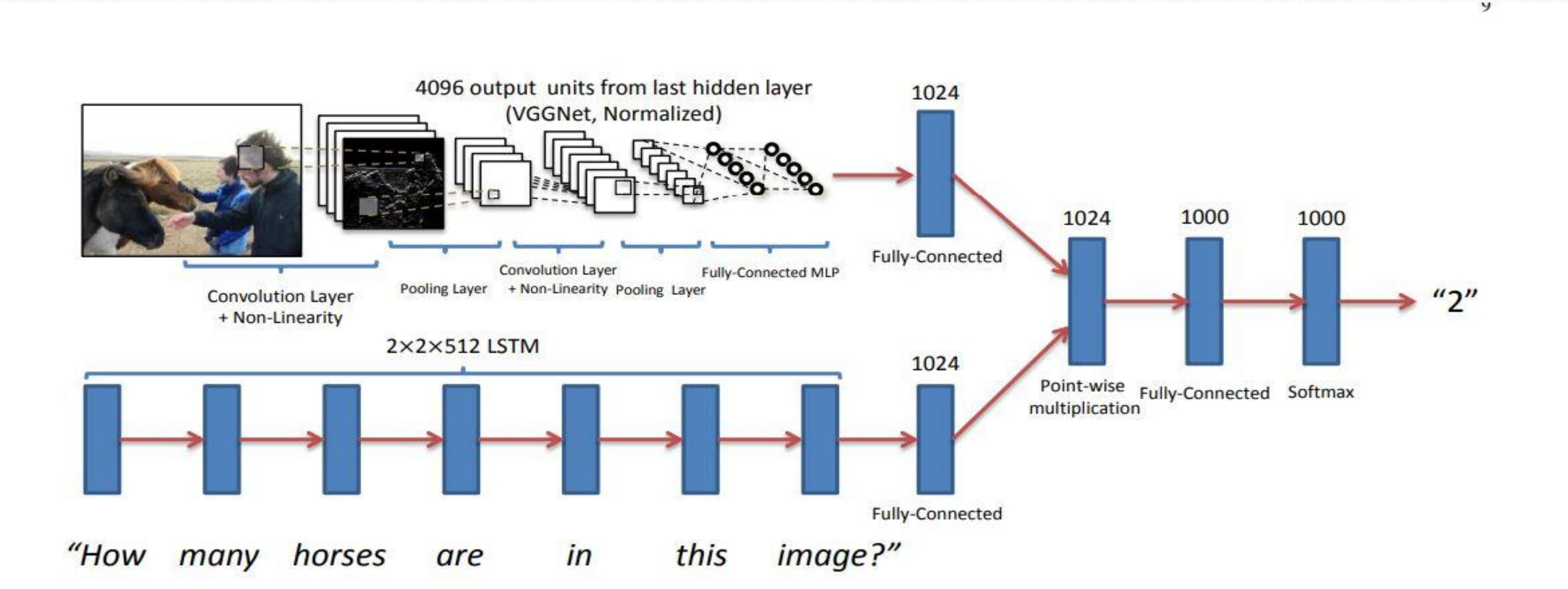
DATASET


Who is wearing glasses?
man
woman
Where is the child sitting?
fridge
arms
Is the umbrella upside down?
yes
no
How many children are in the bed?
2
1

Parameters

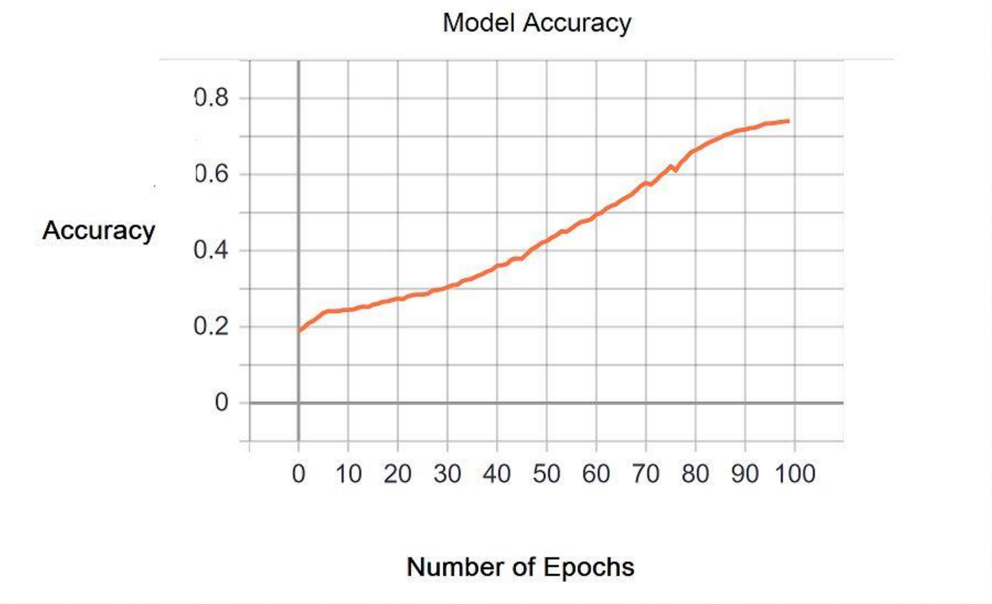
- 50000 Images
- 150000 Questions
- 1000 Possible Answers

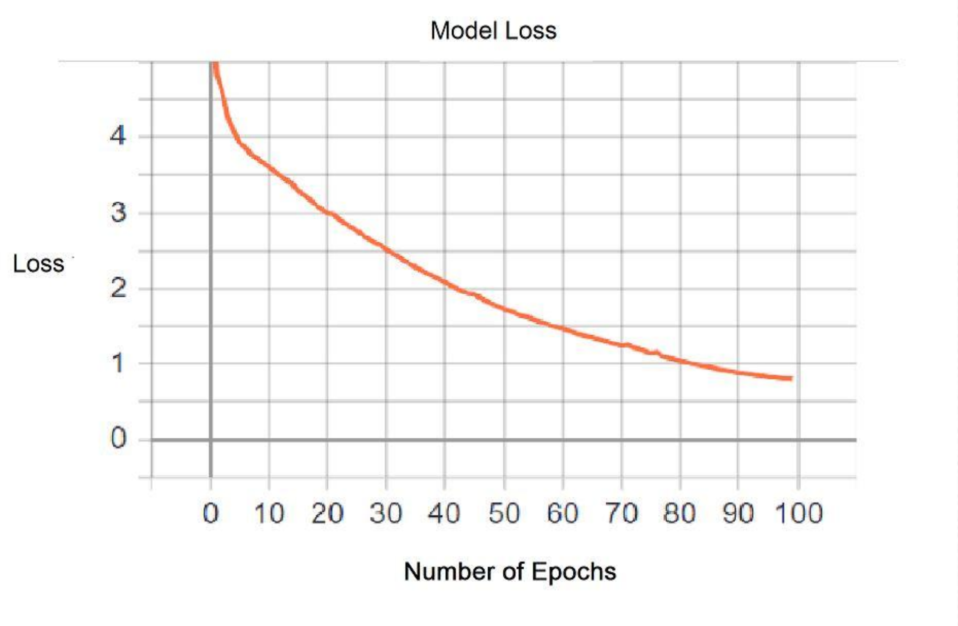
Implementation- Real Time VQA



Implementation- Real Time VQA

Results





Implementation- Floodnet VQA

Datase

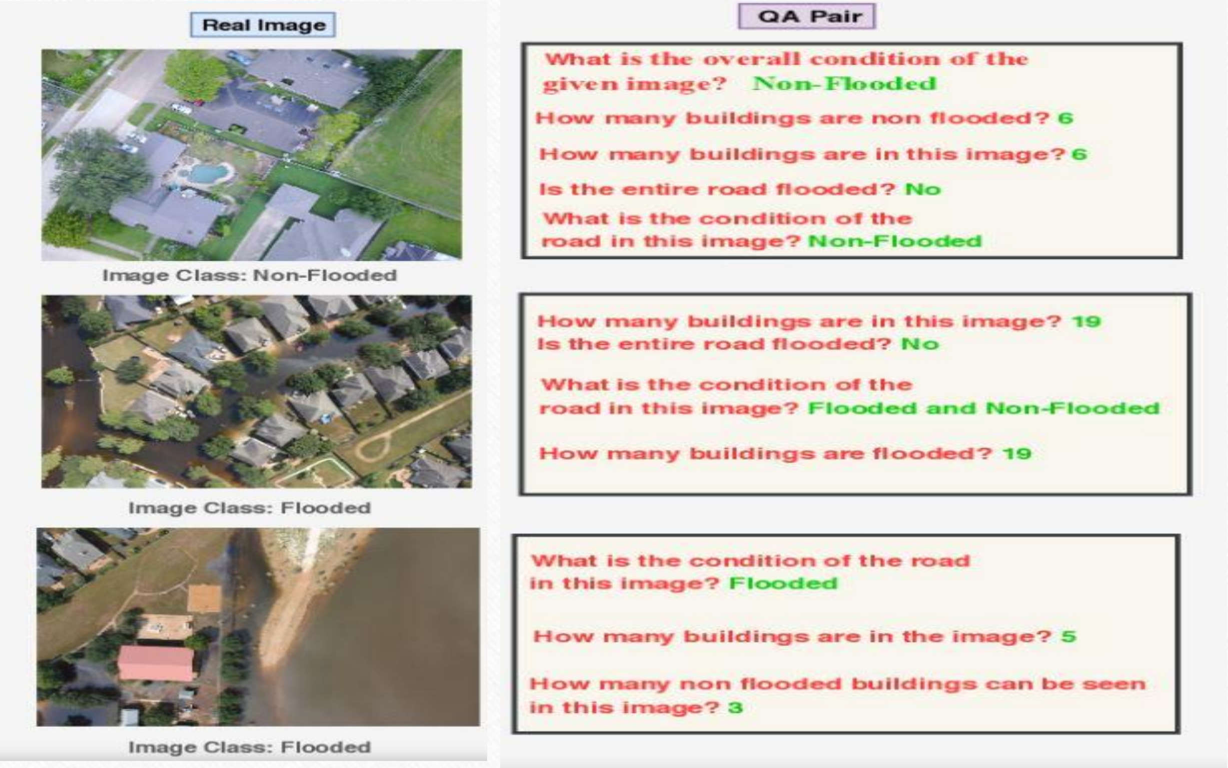


Image Class: Non-Flooded

What is the overall condition of the given image? Non-Flooded

How many buildings are non flooded? 6

How many buildings are in this image? 6

Is the entire road flooded? No

What is the condition of the road in this image? Non-Flooded

Image Class: Flooded

How many buildings are in this image? 19

Is the entire road flooded? No

What is the condition of the road in this image? Flooded and Non-Flooded

How many buildings are flooded? 19

Image Class: Flooded

What is the condition of the road in this image? Flooded

How many buildings are in the image? 5

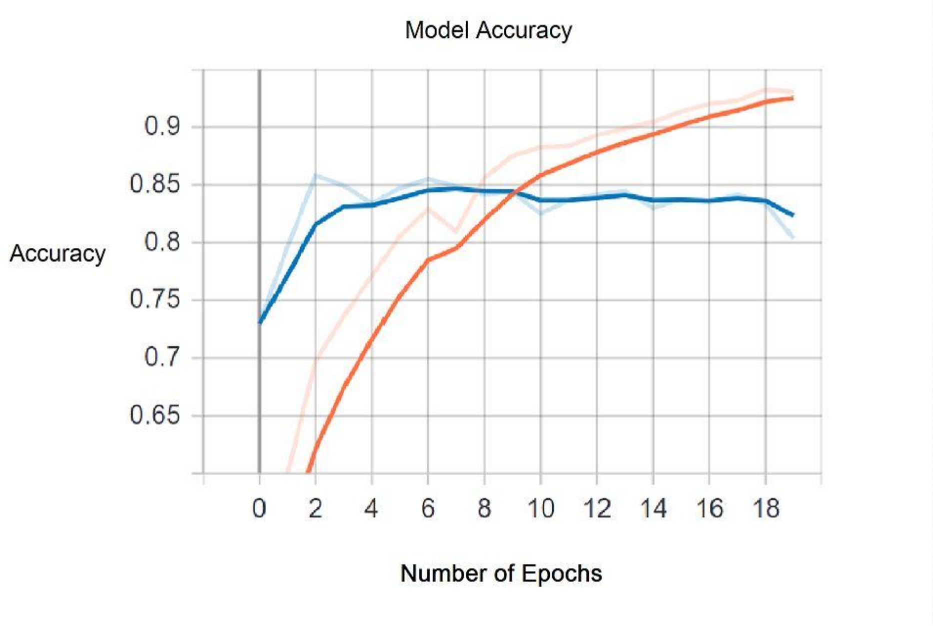
How many non flooded buildings can be seen in this image? 3

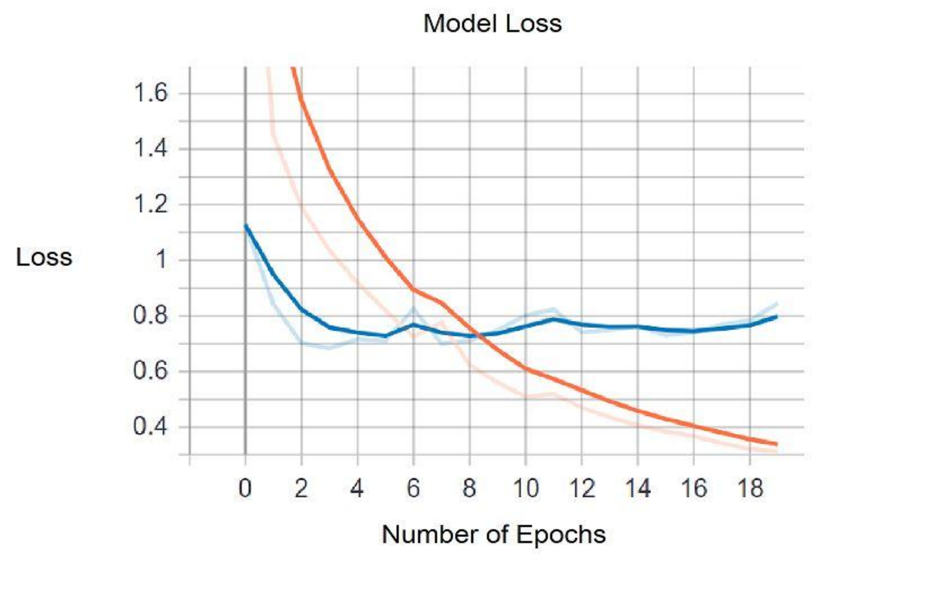
Parameters

- 1450 Images
- 4511 Questions
- Questions - Simple counting,complex counting,condition recognition,yes/no

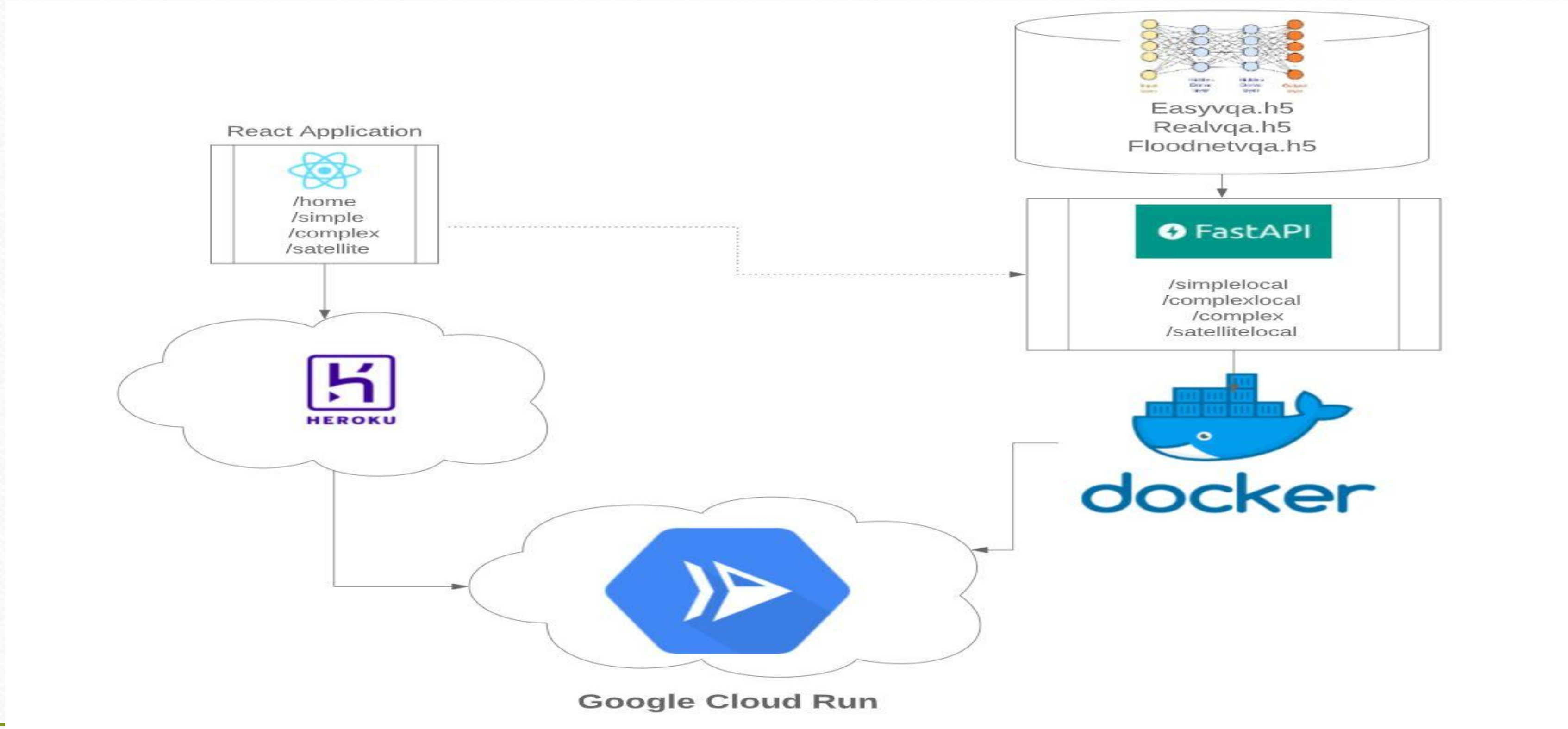
Implementation- Floodnet VQA

Results





Software Architecture



React Application

/home
/simple
/complex
/satellite

Easyvqa.h5
Realvqa.h5
Floodnetvqa.h5

FastAPI

/simplelocal
/complexlocal
/complex
/satellitelocal

HEROKU

docker

Google Cloud Run

THANK YOU