Demo Plan

Image Caption Generation:

Concept: The process for the image captioning generator system is that it is the process of recognising the context of an image and analysing it with given captions. This process is by using natural language processing and computer vision and deep learning. It includes the labelling of an image with English keywords with the help of datasets provided during the training process of the model.

Medical application: 3 embeddings

|  |  |  |
| --- | --- | --- |
| Demo 1 – MedViLL Application | Concept | 3 embeddings (Visual embedding, Joint embedding, Language embedding) |
| Data Validation | The dataset was found from GitHub and it consisted of images of x rays that were clean and already pre-processed before training and it had labels showing each patients’ diagnosis of their organs |
| Build 1 | In the first build we looked at the datasets and how it worked and managed to download the pre-trained weights and the dataset through anaconda command prompt. However, we were facing issues with the importing of certain libraries due to it not being downloaded. Only a few epochs were used when training, however we faced errors in between the training. |
| Build 2 | In the second build, we managed to download and import the files, however, whenever we used the command “python main.py”, it would not allow us to use it due to the missing directory. The directory was used from another pc, and we did not have the same directory which made it hard for us to use that specific command. Due to the main reason for the MIMIC-CXR requiring for us to show the certification, we could not process the whole code |
| Results | We could not get any results due to the missing dataset which lead us to errors in downloading datasets and importing them onto the code. |
| Validation of Effectiveness | Not very effective as we were missing a dataset which was the MIMIC-CXR dataset. The only way for it to be downloaded was through a certification which we did not have therefore we have to look for any other similar dataset and code |
| Demo 2 – Image Captioning Generation | Concept | Simple image captioning generation system which can provide us a caption of a specific image provided to the system |
| Data Validation | When looking at the code and the images, we had to see how the code works for us to know which types of images can be imported in for analysis. We also have to make sure that the code is cleaned and pre-processed so that its ready to be trained and tested |
| Build 1 | We first had a code which could analyse any image that was provided and it would provide us with a caption showing us what is included in the image. However, the code we used before did not work due to the incorrect path and it was quite complex for us to fix |
| Build 2 | In our second build, we looked at a YouTube tutorial on how to create an image captioning system and we made a simple image captioning system where the same concept applied onto the code and it gave us captions from those images provided. No epochs were used in this code |
| Results | Once we ran the code with the images given, some of the captions were almost accurate and some of the images we provided, gave us not so accurate results which tells us that the code itself is not very developed properly as the main thing we are trying to get is accurate results |
| Validation of Effectiveness | Somewhat effective as it can give us a bit of detail of the caption provided. However, the caption provided is not accurate as it falsely identifies the image for something else. |
| Demo 3 – Image Captioning/ Video Surveillance | Concept | A detailed captioning system which can provide us with detailed captions of a specific image |
| Data Validation | The dataset provided was first cleaned and pre-processed before it ready to be used in the code so that we could get accurate results. The dataset consisted of different images. |
| Build 1 | In the first build there were 10 epochs used when training the model and the batch size was 16. These were first used to train the model so that when the code is executed, the model can then give us an accurate result from the images provided |
| Build 2 | The first build gave us somewhat accurate results. But the more epochs used in the model and the bigger the batch size, the more of chance of better accuracy and analyses of the image will be quicker. |
| Results | We managed to get a lot of captions from the images that were given and they were very accurate to the image which indicates that the code Is well developed and can be used to analyse every single image given. It would also give a predicted caption showing the predicted caption of the image and then the actual caption of the image. |
| Validation of Effectiveness | Very effective as the results shown in the captions are the results we have been looking for and it can be used for analysis. This analysis can be used in the medical domain for analysis and provide us with accurate results therefore, better treatment. |  |  |