# Readme

# This is a Image captioning model I have made for week 4 of CSOC, as my final project. This model generates a story on the by extracting the features of an input Image[¶](https://kkb-production.jupyter-proxy.kaggle.net/static/dist/jupyterlab/jupyterlab-index-560be39f31dae074b6dc.html?session=eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0.#This-model-generates-a-story-on-the-by-extracting-the-features-of-an-input-Image)

# So basically, it consists of 4 submodels in it:

# Feature extraction model by extracting a layer from a image classification model (Computer Vision)

# VGG model, used also for feature extraction, using transfer learning(Computer Vision)

# 3. LSTM model, trained on a dataset with predefined captions (Natural Language Processing)

# 4. A Top-p (nucleus) sampling model using transformers, which generate a story from the captions generated (Natural Language Processing)

# I have done this in 3 files. In the first model, “features.ipynb”, I have trained an Image classifier from scratch using RESNET architecture, and then downloaded the extracted features for all images in the dataset in "custom\_resnet\_features.pkl", which is then passed into the next model, in the next and final jupyter notebook, “nlp.ipynb”, where other 2 models are trained, and the final output is shown.

# The first model works properly, classifying the test image with around 70% accuracy into 100 classes. I used the CIFAR-100 dataset for this. However, it seems like this model wasn’t deep enough, or couldn’t be trained deep enough from scratch with not transfer learning using a laptops resources, and thus, the features extracted cant generate the proper captions.

# So, without changing a code, I have used transfer learning from keras’ pretrained VGG16 network, in the jupyter notebook “VGG\_model.ipynb” and applied the features extracted to the same code, and achieved relatively good results, as the model is able to make a story from the image with a reasonable accuracy in the file “nlp.ipynb”..

# The accuracy of the captions would be a lot better if the LSTM model was able to train for more epochs, as right now I was only able to train it for 10 epochs.