Natural Language Understanding

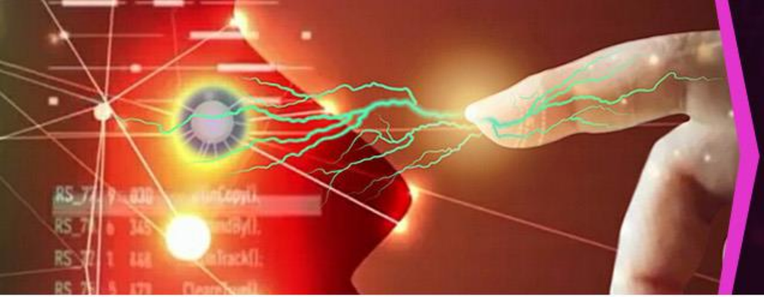
Figurative Languages Caption

Supervised by

Dr: Hamada Nail
Eng: Rowayda Adel

Figurative Languages Caption

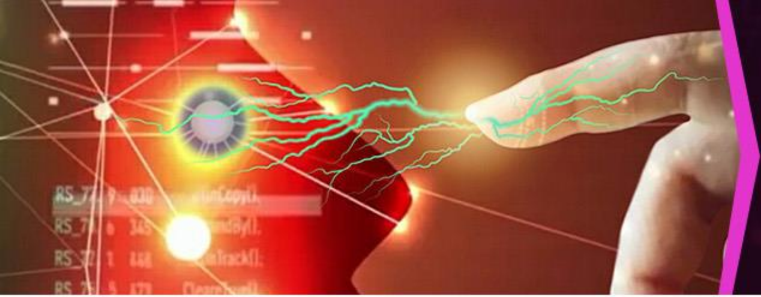
- Team Leader
  - Hazem Refai Mohamed Refai.
- Team Members
  - Khaled Ahmed Slama Eldifrawy.
  - Ibrahim Mousa El Sayed Habib.
  - Eslam Khaled Soliman Gad.
  - Ahmed Samer Eid Abdulhamid.
  - Radwa Walied Rabie Gad.
  - Rawda Walied El Bassiouni.



Figurative Languages Caption

❑ Agenda

- Project Idea.
- Project Goals.
- Project Structure.
- Dataset Description.
- Import libraries
- Preprocessing.
- How to build DL.
- Test Model.



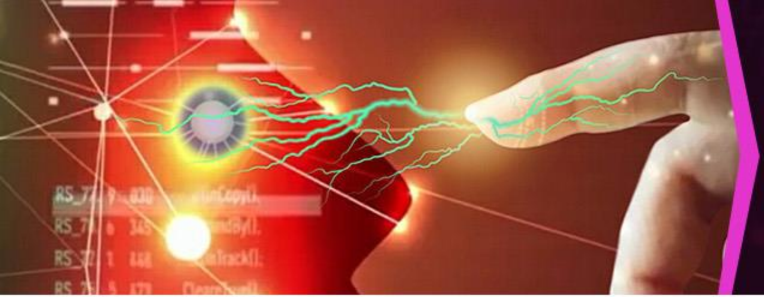
Figurative Languages Caption

❑ Project Idea…?

- **Figurative Languages Caption,** We have a set of data that contains many images and an actual many format caption for each image of them, and all we must do is devise a new caption for each image of them in the testing phase from the old captions.

❑ The Project Goal:-

- **Figurative Languages Caption,** The main goal of this project is to create a caption for a complete description of the image that is entered through the user, and which has a previous description on which the model has been trained before, and it creates a new caption exactly like the **Abstract Summarization.**

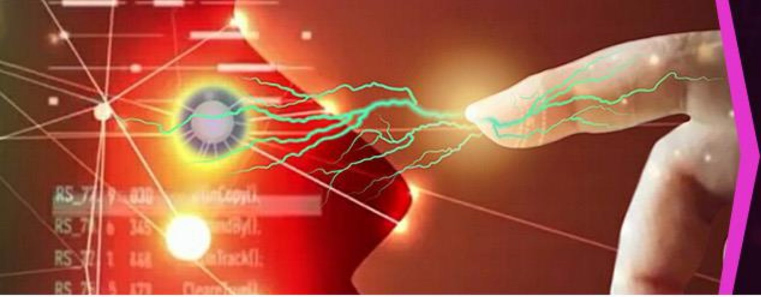


Figurative Languages Caption

❑ Dataset Description…

– The dataset consists of two parts:-

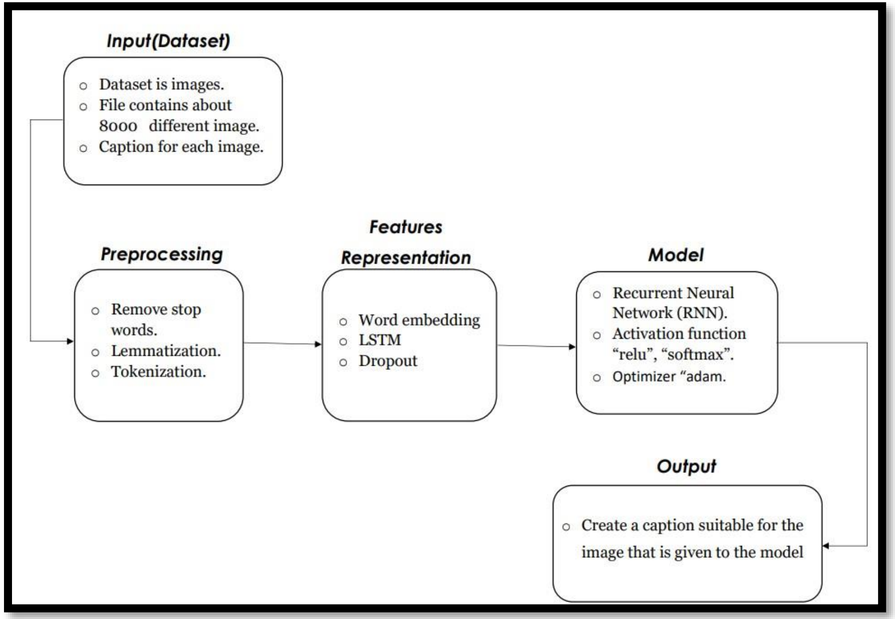
- Images : The images are multiple and different, and they are of type RGP, and they are ready so that the deep learning model can extract features from them easily, and through them we can caption the image
- The Captions: The captions is about texts, and these texts refer what each image contains, and each image has more than one captions in an average of three to five captions for each image, after learning which we can devise a new captions of the image that is compatible with it, and this is done by **NLU**.



Figurative Languages Caption

❑ Project Structure…

- This is the Roadmap that was relied upon to create the project. **First**, we searched for the dataset. **Second**, we did some preprocessing on the text, as well as on the images, in order to extract the features.
- And after we have created a deep learning model and it has been well trained, images will be inserted to test the model and how to create the captions.



Input(Dataset)

- Dataset is images.
- File contains about 8000 different image.
- Caption for each image.

Preprocessing

- Remove stop words.
- Lemmatization.
- Tokenization.

Features Representation

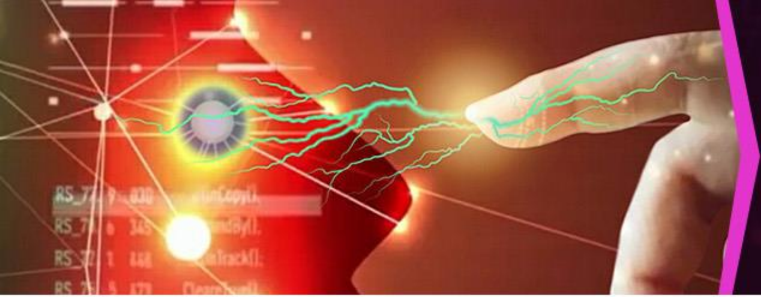
- Word embedding
- LSTM
- Dropout

Model

- Recurrent Neural Network (RNN).
- Activation function “relu”, “softmax”.
- Optimizer “adam.

Output

- Create a caption suitable for the image that is given to the model

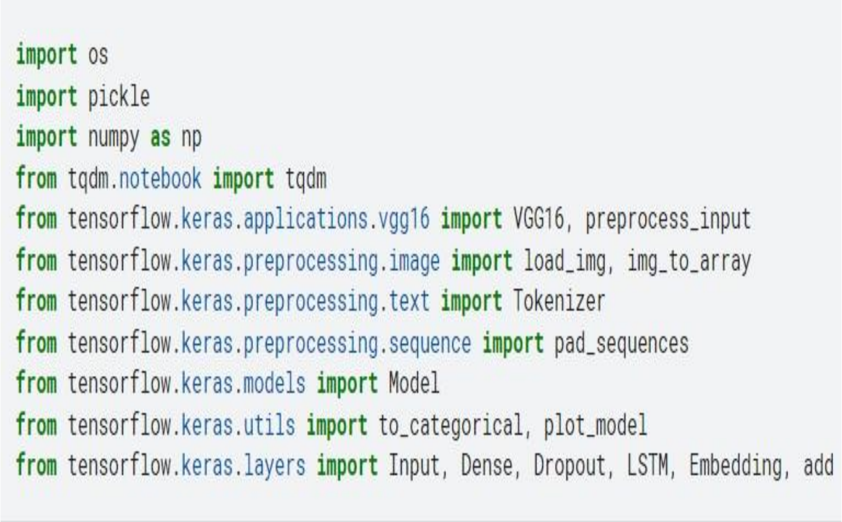


Figurative Languages Caption

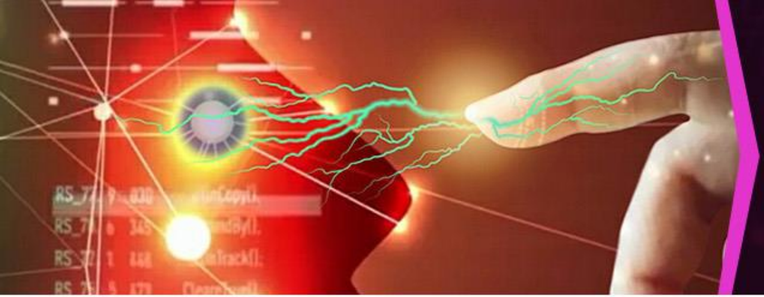
❑ Import libraries…

The most important libraries supporting the project

1. Vgg(16) pretrained model **CNN** for extract feature.
2. pickle used to save feature and redo use again.
3. Keras used to call some function to preprocessing text and image, build model through it and show architecture of our model.
4. Tqdm used in for loop to show execution for processes.



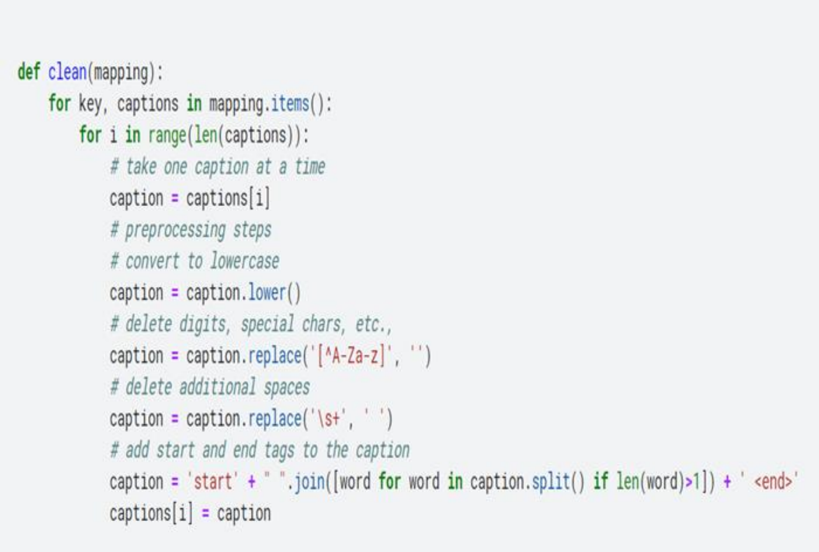
```
import os
import pickle
import numpy as np
from tqdm.notebook import tqdm
from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Model
from tensorflow.keras.utils import to_categorical, plot_model
from tensorflow.keras.layers import Input, Dense, Dropout, LSTM, Embedding, add
```



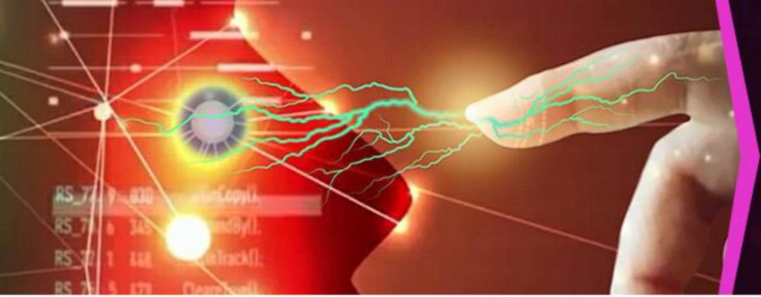
Figurative Languages Caption

❑ Preprocessing….

Since we are studying **NLU** , and this project aims to do so, and this must be clarified in it, it is important for the pre-processing of texts as shown in front of us: Pre-processing is about cleaning the texts that we will use in the training phase from all stop words and special characters, converting the words in the texts to lowercase letters and replacing the (\s) to a space and dividing the sentence is also important in the pre-processing process so that the pure text enters the form during training



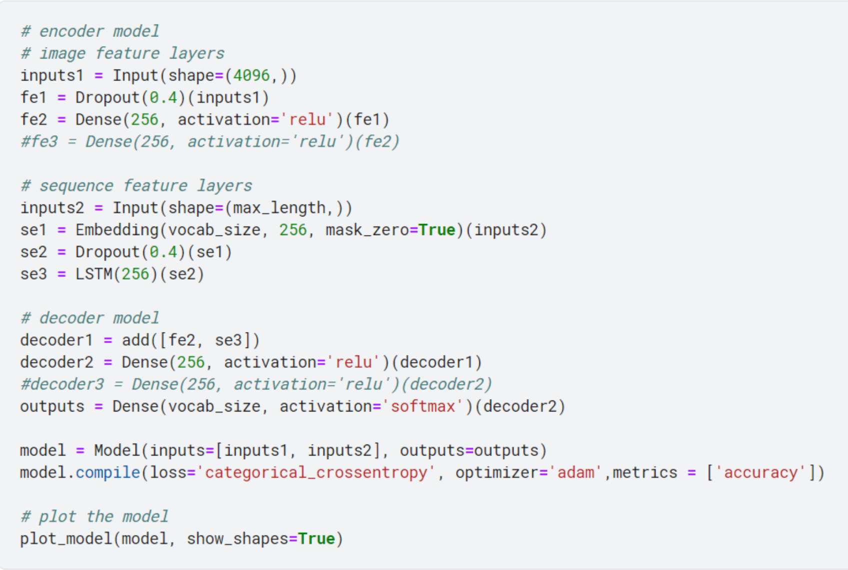
```python
def clean(mapping):
    for key, captions in mapping.items():
        for i in range(len(captions)):
            # take one caption at a time
            caption = captions[i]
            # preprocessing steps
            # convert to lowercase
            caption = caption.lower()
            # delete digits, special chars, etc.,
            caption = caption.replace('[^A-Za-z]', '')
            # delete additional spaces
            caption = caption.replace('\s+', ' ')
            # add start and end tags to the caption
            caption = 'start' + " ".join([word for word in caption.split() if len(word)>1]) + ' <end>'
            captions[i] = caption
```



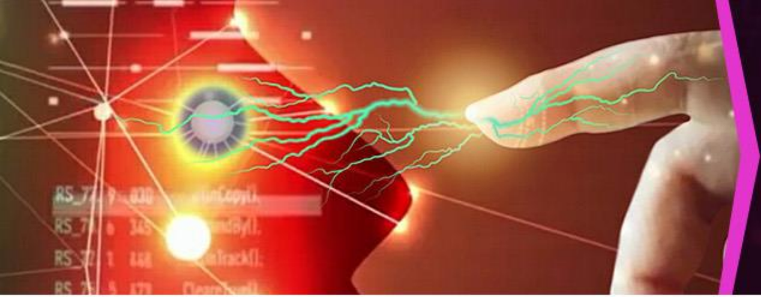
Figurative Languages Caption

❑ How to build DL.

A deep learning model is a model that consists of a neural network that contains input values, some hidden layers, and an activation function that can be added to any of the layers, including the output layers, and the most common, as shown in our project, is that we use the RELU mediation function with hidden layers. Output Layer We use SoftMax and also LSTM Long short-term memory (LSTM) is an artificial neural network used in the fields of artificial intelligence and deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections.



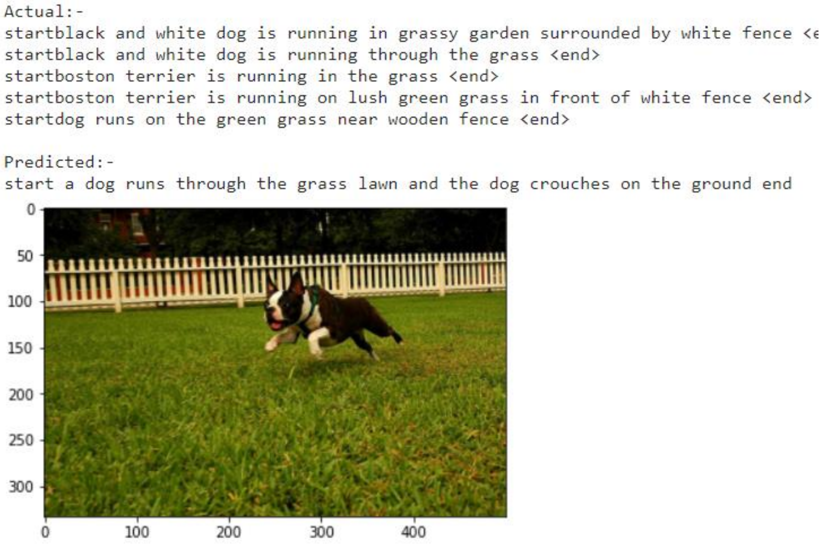
```
# encoder model
# image feature layers
inputs1 = Input(shape=(4096,))
fe1 = Dropout(0.4)(inputs1)
fe2 = Dense(256, activation='relu')(fe1)
#fe3 = Dense(256, activation='relu')(fe2)

# sequence feature layers
inputs2 = Input(shape=(max_length,))
se1 = Embedding(vocab_size, 256, mask_zero=True)(inputs2)
se2 = Dropout(0.4)(se1)
se3 = LSTM(256)(se2)

# decoder model
decoder1 = add([fe2, se3])
decoder2 = Dense(256, activation='relu')(decoder1)
#decoder3 = Dense(256, activation='relu')(decoder2)
outputs = Dense(vocab_size, activation='softmax')(decoder2)

model = Model(inputs=[inputs1, inputs2], outputs=outputs)
model.compile(loss='categorical_crossentropy', optimizer='adam',metrics = ['accuracy'])

# plot the model
plot_model(model, show_shapes=True)
```



Figurative Languages Caption

❑Test Model.

In the testing phase, the data set is divided into a part for training and another part for testing, and an image is entered by the user into the model, and thus the model automatically extracts the features and determines the description that fits with these features through the description that was previously trained during the training period.



```
Actual:-
startblack and white dog is running in grassy garden surrounded by white fence <e
startblack and white dog is running through the grass <end>
startboston terrier is running in the grass <end>
startboston terrier is running on lush green grass in front of white fence <end>
startdog runs on the green grass near wooden fence <end>

Predicted:-
start a dog runs through the grass lawn and the dog crouches on the ground end
```

THANK YOU