Deep Learning Course

Satellite Image Classification

JOHN KOUMENTIS, SOTIRIS PANOPOULOS

Scope

The project evaluates the performance of the encoder - decoder with attention mechanism architecture, over predicting the correct set of labels of the given satellite image chip.

Data Acquisition

The JPG version of the satellite images dataset *Planet: Understanding the Amazon from Space** was used.

The screenshots are chips extracted from the bigger dataset that are provided as a reference to the scene content.

* https://www.kaggle.com/c/planet-understanding-the-amazon-from-space

Data Preprocessing

- Dictionary that maps each image path with its labels:

```
['<start> agriculture habitation partly_cloudy primary road water <end>']
```

- Feature extraction
  - Pretrained ResNet50_V2.
  - Features from last convolutional layer are extracted and cached ((7 * 7 * 2048) floats per image).
- Tokenize labels
  - Labels are tokenized by getting split on spaces.
  - Word to index and index to word mapping.
  - Padding is applied, each created sequence must be of the same length as the longest one.
- Spit to training and validation sets
  - 80% and 20% of the examples respectively.

Model

- Model architecture is inspired by the Show, Attend and Tell* paper. The authors propose an attention based model that automatically learns to describe the contents of images.
- We extract the features from the lower convolutional layer of ResNet50_v2 giving us a vector of shape (7, 7, 2048) which is flattened to a shape of (49, 2048).
- This vector is passed through the CNN encoder.

```
Model: "cnn_encoder"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                multiple                  524544
=================================================================
Total params: 524,544
Trainable params: 524,544
Non-trainable params: 0
_________________________________________________________________
```

* https://arxiv.org/pdf/1502.03044.pdf

Model

- The encoder input is passed through the RNN decoder, that attends over the image to predict the next word.

- The used attention mechanism is based on Bahdanau's additive attention**. This frees the model from having to encode the whole input feature into a fixed-length vector, and lets the model focus only on information relevant to the generation of the next target word.

```
Model: "rnn__decoder"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        multiple                  115200
_________________________________________________________________
gru (GRU)                    multiple                  1575936
_________________________________________________________________
dense_1 (Dense)              multiple                  262656
_________________________________________________________________
dense_2 (Dense)              multiple                  230850
_________________________________________________________________
bahdanau_attention (Bahdanau multiple                  394753
=================================================================
Total params: 2,579,395
Trainable params: 2,579,395
Non-trainable params: 0
_________________________________________________________________
```

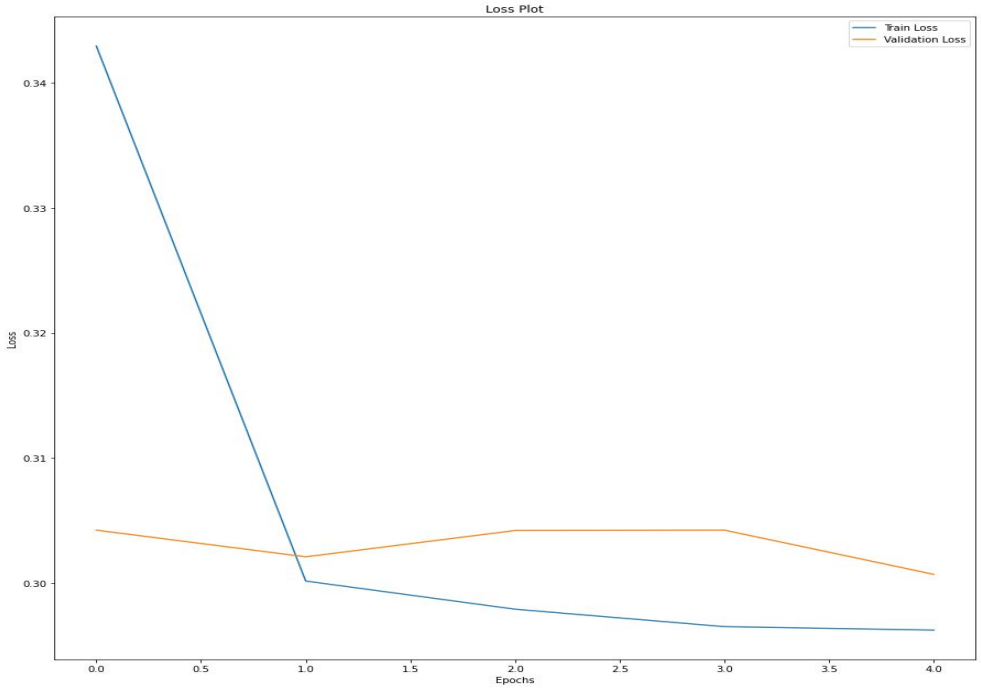
** https://arxiv.org/pdf/1409.0473.pdf

Training pipeline

- The extracted features stored in the respective .npy files are passed through the CNN encoder.
- The encoder output, hidden state (initialized to 0) and the decoder input (which is the start token) is passed to the RNN decoder.
- The decoder returns the predictions and the decoder hidden state.
- The decoder hidden state is then passed back into the model and the predictions are used to calculate the loss.
- Teacher forcing is used, to decide the next input to the decoder.
- Calculate the gradients, apply them to the optimizer and backpropagate.

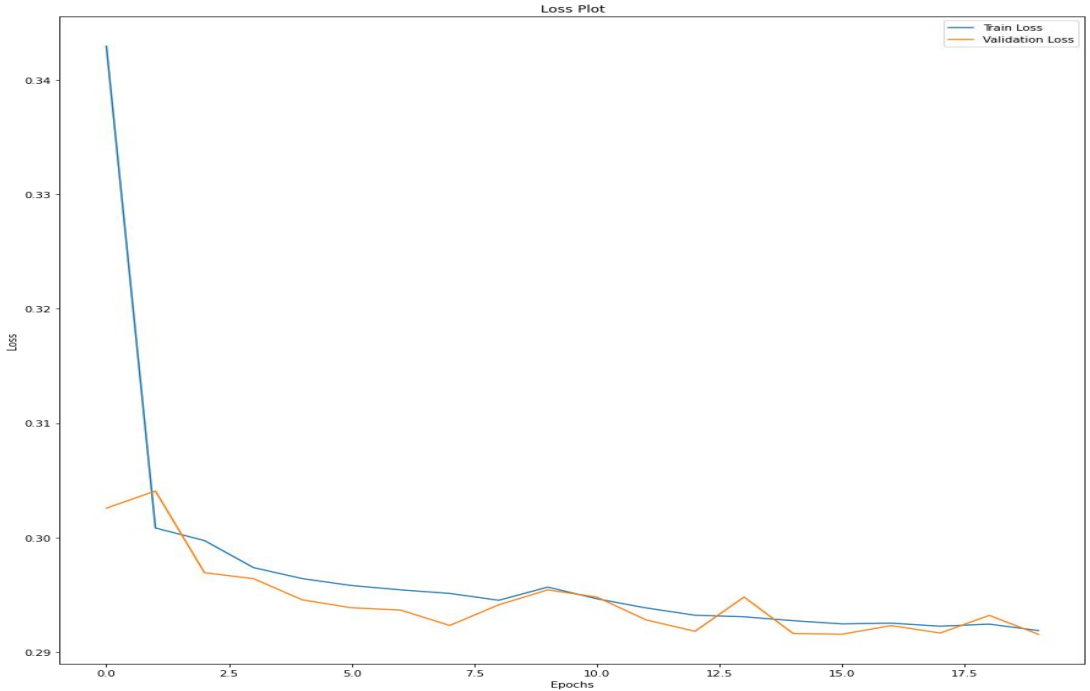
Results

- Model trained for 11 epochs achieved F-beta score 83.39%
- Validation loss reached 0.2962 at 11th epoch, while training loss reached 0.2924
- After the first epoch the validation loss decrease was only occurring at 3rd decimal place



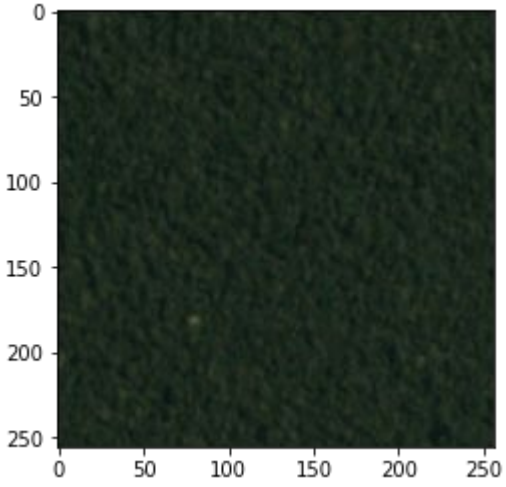
Results

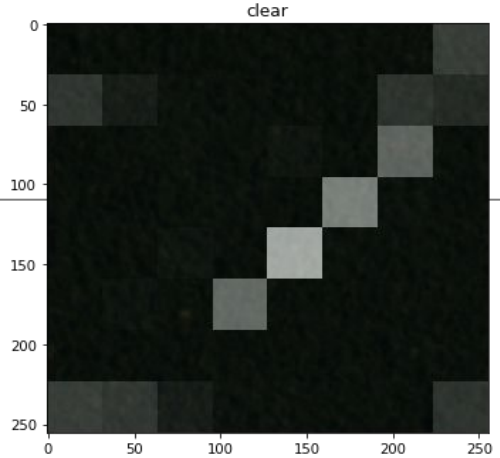
- Model trained for 20 epochs achieved F-beta score 83.9%
- Validation loss reached 0.2915 at 20th epoch, while training loss reached 0.2918
- After the first epoch the validation loss decrease was only occurring at 3rd decimal place

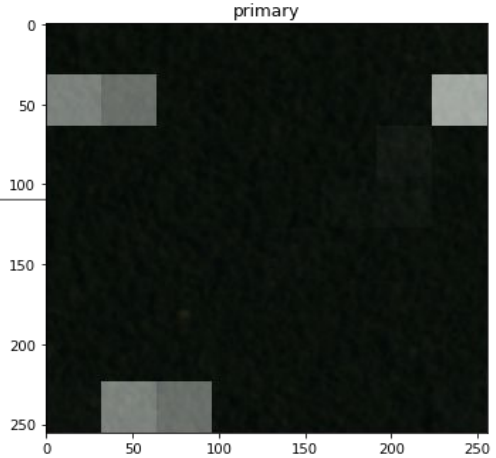


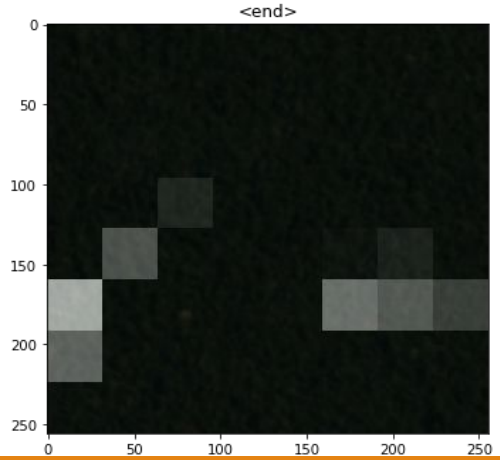
Example

Real: clear, primary



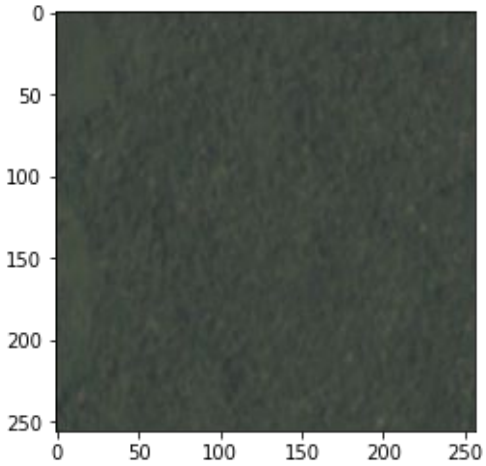


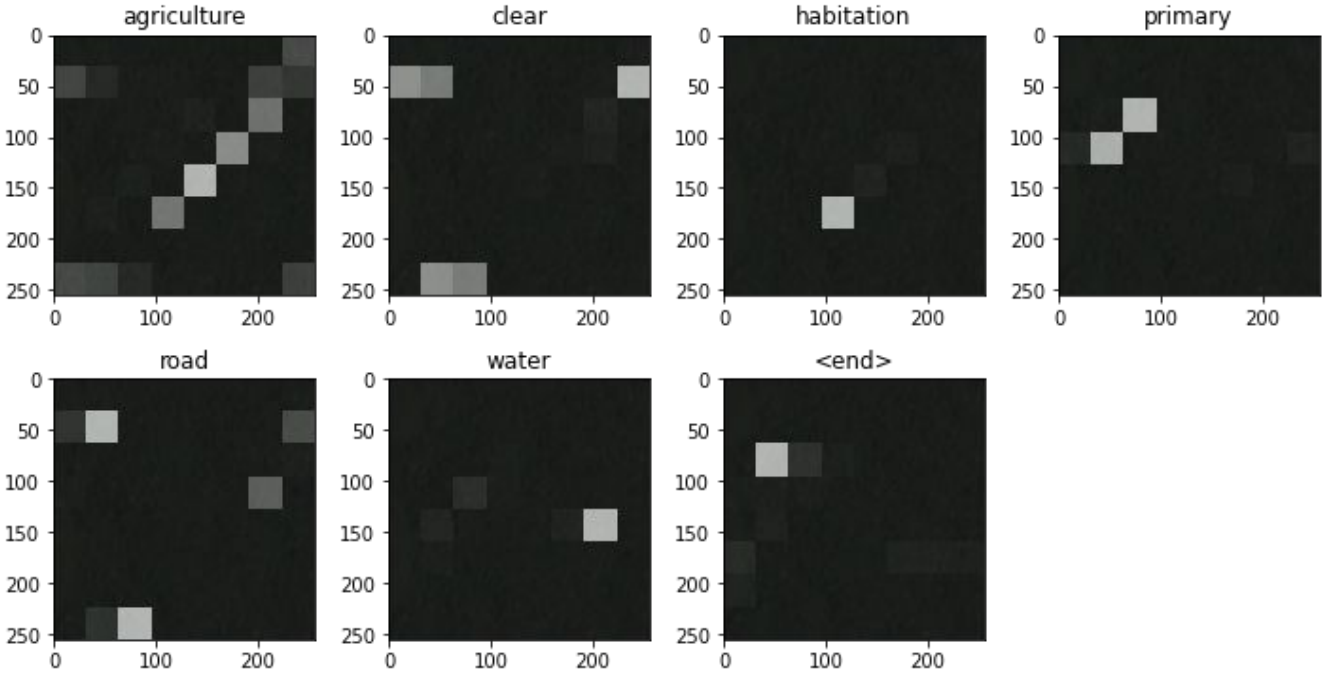




Example

Real: clear, primary





Conclusions

- Satellite image classification using attention mechanism is a feasible approach. The model achieved F-beta score greater than 83% (highest F-beta score from the respective Kaggle competition around 93%).
- An approach that utilizes the TIF version would probably provide better results, however it would be resource demanding.
- We used the ResNet pretrained at the ImageNet dataset, as feature extractor. Probably, fine-tuning the model to the satellite image dataset before proceeding with the feature extraction would provide better results.
- Model training took place inside a GPU enabled Colab notebook (limitations at resources).
- Future improvements include selecting a different pre-trained network as feature extractor, modifying model architecture by replacing the GRU layer with an LSTM one and/or selecting a different attention mechanism.