CDS Visual Analytics Assignment #3: Simple Image Search Algorithm (+KNN)

What is this?

Setup

1. Make sure to have python and Git Bash installed!
2. Open a Git Bash terminal and use Git to download the repository:

```
git clone https://github.com/missingusername/cds-vis-git.git
```

3. Navigate to the project folder for this assignment:

```
cd assignments/assignment3
```

4. Before doing anything else, you first need to get the tobacco dataset. You can download the `Tobacco3482` dataset manually from Kaggle. When downloaded, unzip the `archive.zip` and place the `Tobacco3482-jpg` folder inside the `in` folder of the `assignment3` directoy. However, remember to delete the duplicate `Tobacco3482-jpg` folder inside `Tobacco3482-jpg` itself. This should leave you with a file structure like this:

```
assignment3/
    ├── in/
    │   └── tobacco3482-jpg/
    │       ├── ADVE
    │       ├── Email
    │       ├── Form
    │       ├── Letter
    │       ├── Memo
    │       ├── News
    │       ├── Note
    │       ├── Report
    │       ├── Resume
    │       └── Scientific
    ├── out/
    │   ├── classification_report_with_vgg16.txt
    │   └── learning_curves_with_vgg16.png
    └── src/
        └── main.py
```

5. Before you can run the scripts, make sure you have the required libraries in the `requirements.txt`. This can be done by simply running the OS-appropriate setup script from inside the `assignment3` folder, which will set up a virtual environment and get the required libraries. Again, using Git Bash:
6. To finally execute the script, simply run the OS-appropriate `run.sh` script in the same Git Bash terminal:

Unix:

```
bash unix_run.sh
```

Windows:

```
bash win_run.sh
```

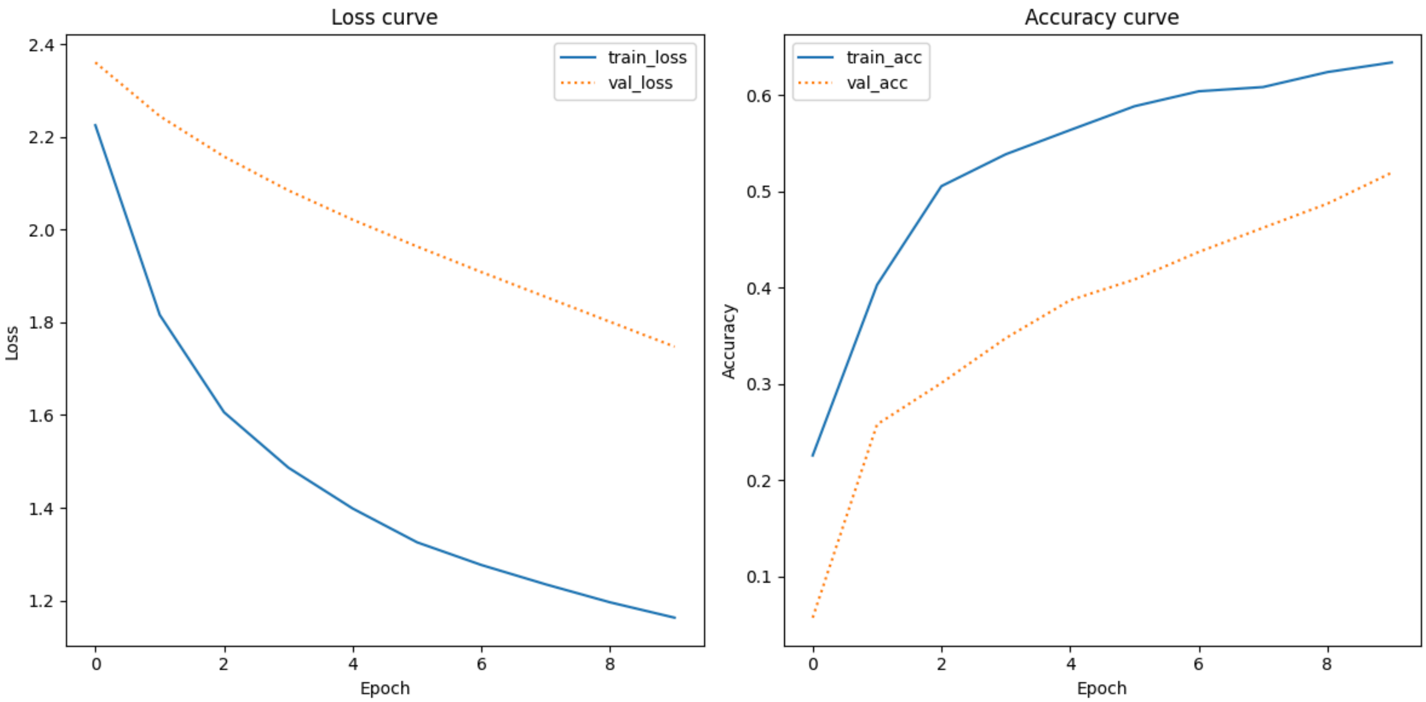
Takeaways from output

Class	Precision	Recall	F1-Score	Support
ADVE	0.92	0.79	0.85	57

Class	Precision	Recall	F1-Score	Support
Email	0.85	0.84	0.84	135
Form	0.42	0.65	0.51	88
Letter	0.45	0.84	0.59	122
Memo	0.39	0.33	0.36	109
News	0.75	0.62	0.68	34
Note	0.58	0.19	0.29	36
Report	1.00	0.06	0.12	48
Resume	0.00	0.00	0.00	15
Scientific	0.47	0.13	0.21	53
Accuracy			0.56	697
Macro Avg	0.58	0.45	0.44	697
Weighted Avg	0.60	0.56	0.53	697

As we can see from the classification report, the final model is actually alright at clssifying different document categories, with a WEIGHT AVG. f-1 score of **0.53**. This means that, on average, the model is good at classifying the different types of documents.
Looking at the specific classes, we can see that it seems especially good at **ADVE** and **Emails**, while seeming less good at **Reports**, and even completely missing **Resumes**.

However, if we look at the final plot of the learning curves, we can see that there is a noticable discrepancy between the lines. In the loss curve, we can see that the val_loss and train_loss seem to start off about similar, but that train_loss drops off significantly faster than the val_loss, with val_loss barely falling at all. This would indicate that the model is somehow overfitting to the training data, and isnt learning how to generalize to unseen data.



Looking at the accuracy curves, we can see that the model quickly becomes good at classifying the training data, while the validation lacks slightly behind, but both still following the same general upwards trend, leading me to believe, that if given more epochs, the curves could converge.

Try as i might, i have not been able to fix these problems of overfitting. i have tried implementing adam, sgd, different learning rates, with and without data augmentation, batchnormalization, dropout layers, different random states, nothing seems to work. So while i realize that the script is somehow overfitting, i cannot find the actual cause no matter what i try.

Limitations and possible steps to improvement

As mentioned, the model seems to somehow overfit, which is of course a great limitation. The steps to improving the model would be to combat this overfitting.