DeepLenses

Common Task: 1 :

Wandb link: https://wandb.ai/gakash2001/deeplenses_task-1?workspace=user-gakash2001

Model	Accuracy(%)	F1 Score	AUC ROC Score
Resnet50	93.73	-	-
EfficientNetB0	93.57	-	-
EfficientNetB3	94.25	0.94	0.99
EfficientNetB4	94.87	-	-
EfficientNetB5	95.37	-	-
EfficientNetB6	95.933	-	-
EfficientNetB7	95.267	0.95	0.99

EDA

- We started with basic exploratory data analysis.
- The class counts show that the classes are no, sphere and vort, which are perfectly balanced, with 10,000 images in each class.
- All the images have a constant shape of 150x150.
- The images are single-channeled, but since most common and popular image classification architectures work on rbg channel, they have been converted to 3 channeled images using cv2.
- The images are stored as NumPy arrays, and numpy is used to read them.

csv_maker

- A csv file usually helps to handle the data in the form of dataframes. Therefore, csv_maker creates the csv for train and test data.

train

- Seed has been set to make the results reproducible.
- CustomDataset class converts the data to required format for both train and test.
- train data is split to 90:10 ratio, with the 90 for training and 10 for validation.

- No augmentations have been used., but simple augmentations would help. The type of augmentations to be used depend on the data, therefore, in-depth analysis of the data and more information about it would help.
- Different models were used beginning from the lower models to ensure that the model is not overfitting with the given data. Overfitting can be reduced by the augmentation techniques. The heavier models, as excepted were performing better.
- Since the task is multi-class classification, and the classes were perfectly balanced, the most basic and common loss function, CrossEntropy is used.
- The adam optimizer is used, since it is the most standard one, but optimizers like Ranger should help more.
- ExponentialLR scheduler has been used, but schedulers like CosineAnnealingwarmrestarts often performs well at number beating and can be used.
- Most experiments have been run for 50 to 100 epoches and the useful ones have been logged on wandb.
- Efficientnet-b6 performed the best, giving an accuracy of 95.9.

Test

- test notebook is used to test the model trained on the training data on the test data. The trained model weights are sent to the test-notebook. It evaluates the model and returns the accuracy, auc_roc score, f1 score and the auc_roc curve.

Specific Task: 2

Wandb Link: https://wandb.ai/gakash2001/deeplenses?workspace=user-gakash2001

Model (F1 score/AUC Score)	Upsampling	WeightedRandomSampling	Upsampling with Augmentations	Upsampling BCE loss with Augmentations
EfficientNetB0	0.87/-	0.86/-	-	0.90/-
EfficientNetB3	0.87/-	0.88/-	-	-
EfficientNetB5	0.90/-	0.88/-	-	-
EfficientNetB6	0.90/-	0.91/-	-	-
EfficientNetB7	-	--	0.93/0.92	0.94/0.93

EDA

- We started with basic exploratory data analysis.

- The class counts show that the classes are is_lens:(0 or 1 for negative and positive respectively}, which are imbalanced(2.3:1) with 12,574 images in positive and 5426 images in negative class.
- All the images have a constant shape of 101x101.
- The data also has additional meta data {Einstein_area, numb_pix_lensed_image, flux_lensed_image_in_sigma**}** which could have been used by extracting the propabilities for each class or the encodings and passing with the normalized meta data through a 2 layer cnn.
- However, the task mentioned that we need to perform image classification using the images, therefore, the extra data wasn't utilized by me.
- The images are single-channeled, but since most common and popular image classification architectures work on rbg channel, they have been converted to 3 channeled images using cv2.
- The images are stored as fits files, and astropy library is used to read them.
- Running the eda notebook automatically fills the data_fit folder with 10 random images fromthe dataset, which you used for analysing the images. They are stored with the name imageid_label.jpg .

train

- Seed has been set to make the results reproducible.
- CustomDataset class converts the data to required format for both train and test.
- train data is split to 90:10 ratio, with the 90 for training and 10 for validation.
- RandomHotizontalFlip, RandomVerticalFlip and RandomRotation(+15 degree to -15 degree) have been used for augmentation. They were determined by a quick glance through the data.
- The type of augmentations to be used depends on the data, therefore, in-depth analysis of the data and more information about it would help determine better, about which augmentation techniques should be used.
- Different models were used beginning from the lower models to ensure that the model is not overfitting with the given data. Overfitting can be reduced by the augmentation techniques. The heavier models, as excepted were performing better.
- Since the task is binary-class classification, and the classes are imbalanced balanced, BCELosswithLogits provided better results. Due to the imbalance, loss functions like smoothingloss and focal loss are expected to further improve the results.

- The adam optimizer is used, since it is the most standard one, but optimizers like Ranger should help moreas it has look ahead property, which can help deal with imbalance.
- ExponentialLR scheduler has been used, but schedulers like CosineAnnealingwarmrestarts often performs well at number beating and can be used.
- Oversamping using imblearn's oversample seemed to perform better than the weighted random sampler and under sampler. Therefore for the final results, upsampler has been used.
- Most experiments have been run for 50 to 100 epoches and the useful ones have been logged on wandb. The heavier models with augmentations needed way higher number of epoches to converge, hence were run for far larger number of epoches.
- Efficientnet-b7 performed the best, giving an accuracy of 0.94 F1-score and 0.93 AUC-ROC score.

Test

- test notebook is used to test the model trained on the training data on the test data. The test data is same as the validation data, and is obtained by using the same split seed The trained model weights are sent to the test-notebook. It evaluates the model and returns the accuracy, auc_roc score, f1 score and the auc_roc curve.

Note:

I am open to work on any of these projects: Lens Finding, Learning Mass of Dark Matter Halo, Exploring Transformers or Self-Supervised Learning. I shall provide the code for the task on the same repository and update the pdf on mail, as soon as possible.
Thank you.
Akash Gupta
gakash2001@gmail.com