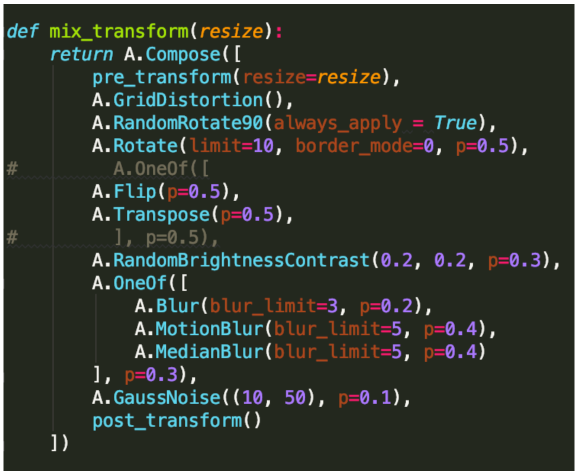
Model documentation and write-up

1. Who are you (mini-bio) and what do you do professionally?
   - Data Science student at Skoltech

   - I am graduated from MIPT with a bachelor degree in Applied Math and Physics in 2018. For over 4 years I have worked in several companies as a Data Scientist where I was involved in almost all types of ML problems — classic ML, CV and NLP. Also I really like competitions and I am the Kaggle former 6th.

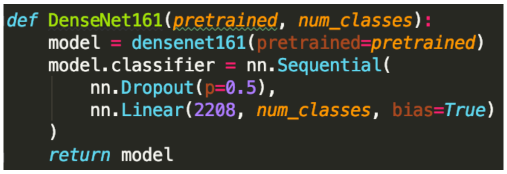
2. High level summary of your approach: what did you do and why?
   I trained a few models for image classification. Basically I thought that some standard approach should perform well. So we used a multiple models and then blended the predictions

3. Copy and paste the 3 most impactful parts of your code and explain what each does and how it helped your model.



```python
def mix_transform(resize):
    return A.Compose([
        pre_transform(resize=resize),
        A.GridDistortion(),
        A.RandomRotate90(always_apply = True),
        A.Rotate(limit=10, border_mode=0, p=0.5),
#          A.OneOf([
        A.Flip(p=0.5),
        A.Transpose(p=0.5),
#          ], p=0.5),
        A.RandomBrightnessContrast(0.2, 0.2, p=0.3),
        A.OneOf([
            A.Blur(blur_limit=3, p=0.2),
            A.MotionBlur(blur_limit=5, p=0.4),
            A.MedianBlur(blur_limit=5, p=0.4)
        ], p=0.3),
        A.GaussNoise((10, 50), p=0.1),
        post_transform()
    ])
```

Good image augmentations were pretty useful as usual (I made a few experiments with different compositions of augmentations).



```python
def DenseNet161(pretrained, num_classes):
    model = densenet161(pretrained=pretrained)
    model.classifier = nn.Sequential(
        nn.Dropout(p=0.5),
        nn.Linear(2208, num_classes, bias=True)
    )
    return model
```

Adding DropOut to DenseNet161 increased its performance for this problem

4. What are some other things you tried that didn't necessarily make it into the final workflow (quick overview)?

Adding clipping for predictions might be useful for this type of metric, but it actually did not increase out score, so we did not use any clipping finally

5. Did you use any tools for data preparation or exploratory data analysis that aren't listed in your code submission?
   No

6. How did you evaluate performance of the model other than the provided metric, if at all?
   Only given metric

7. Anything we should watch out for or be aware of in using your model (e.g. code quirks, memory requirements, numerical stability issues, etc.)?
   Basically speaking the pipeline considers all the staff. I would recommend not to change the batch size for reproducibility

8. Do you have any useful charts, graphs, or visualizations from the process?
   No specific visualizations were made

9. If you were to continue working on this problem for the next year, what methods or techniques might you try in order to build on your work so far? Are there other fields or features you felt would have been very helpful to have?
   Probably specific models for distinguishing similar classes