

# IEEE's Signal Processing Society Camera Model Identification

[ods.ai] GPU\_muscles\_SPcup\_eligible

**Ilya Kibardin**

@ikibardin

**Artur Fattakhov**

@fartuk1



Artur Kuzin

@n01z3



Valeriy Babushkin

@venheads



Andrey Kiselev

@akiselev

# Problem statement

Classes: 2x



7x



1x



Train

10x 275



Full images  
Center crops,  
no changes

Test

10x 132  
Altered

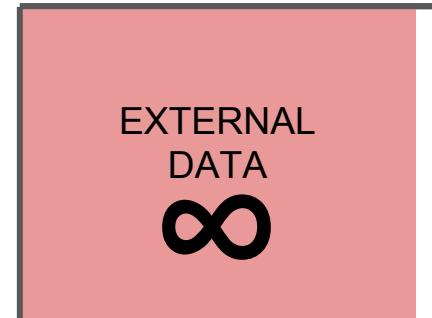
10x 132  
Manipulated

512 x 512

Center crops, no changes

JPEG compression (70, 90)  
Resize (0.5, 0.8, 1.5, 2.0)  
Gamma correction (0.8, 1.2)

EXTERNAL  
DATA  
 $\infty$



# Main idea

Суть™

Shot on

## DSLR camera:

### Cats



### Dogs



Shot on

## Phone camera:



# tl;dr.py

```
from internet import yandex_fotki, flickr, wiki_commons
from Andres_Torrubia import Ivan_Romanov as pytorch_baseline
import kaggle

dataset = kaggle.data()
for source in [yandex_fotki, flickr, wiki_commons]:
    dataset[train].append(source.download())

predicts = []
for model in [densenet201, resnext101, se_resnext50, dpn98
              densenet161, resnext101_d4, se_resnet50, dpn92]:

    with pytorch_baseline():
        model.fit(dataset[train])

    predicts.append(model.predict_tta(dataset[test]))

kaggle.submit(gmean(predicts))
```

# Merge decision for student's second stage eligibility



n01z3

Leader

Mentor with Hardware  
(as graduate student)



ValeriyBabushkin

One faculty member



Artur Fattakhov (MIPT DIHT)

First undergraduate student



Ilya Kibardin

Second undergraduate student



Andrey Kiselev

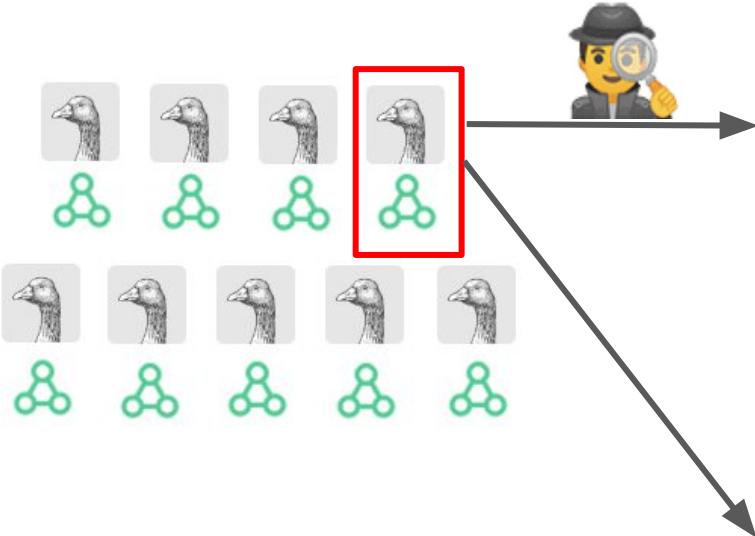
Third undergraduate student

# SPcup\_eligible

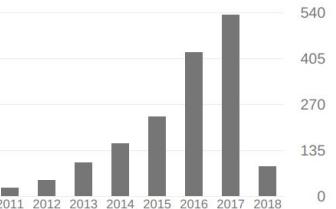
Smoker



Healthy person



Citations	1810	1538
h-index	26	22
i10-index	43	39



Luisa Verdoliva

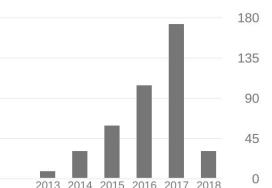
University Federico II of Naples

Verified email at unina.it - [Homepage](#)

Image Processing Multimedia Forensics



Citations	410	407
h-index	11	11
i10-index	12	12



Davide Cozzolino

University Federico II of Naples

Verified email at unina.it

Image Processing



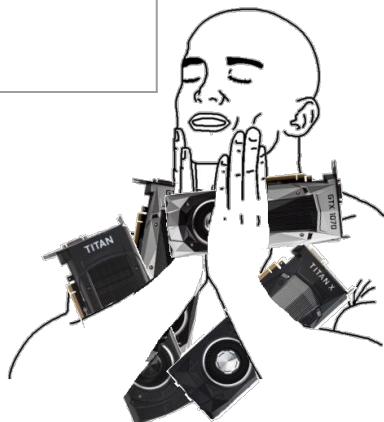
# Hardware

n01z3-dl1	i7-6700K	32Gb	2x Titan X Maxwell
n01z3-dl2	Xeon 2696v3	64Gb	4x 1080Ti
n01z3-dl3	i7-5930K	32Gb	3x 1080Ti
n01z3-home	i7-7700K	64Gb	2x 1080
ilya-home	i7-3770k	16gb	2x 1080Ti

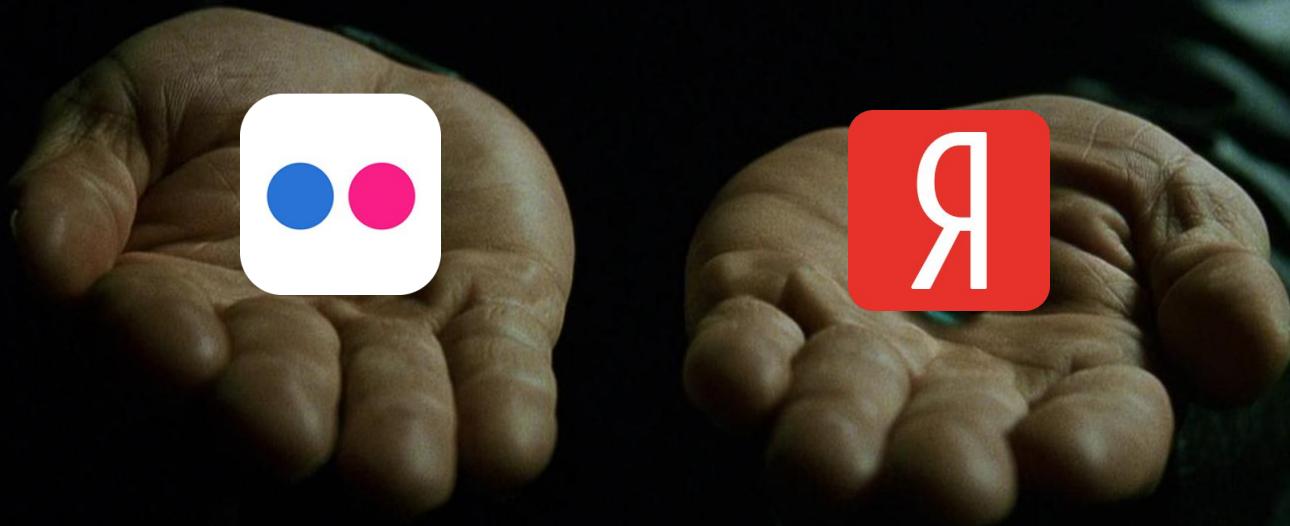
~ 0.11 BTC / month  
(~1100 \$)

vs

Team prize:  
8000 \$



# Sources of external data: how to choose?

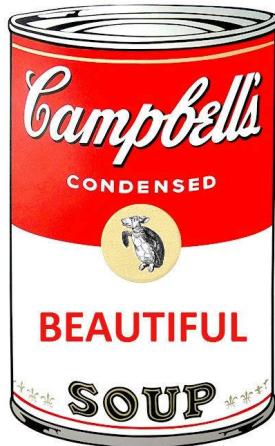


We take all : 500+ Gb photos

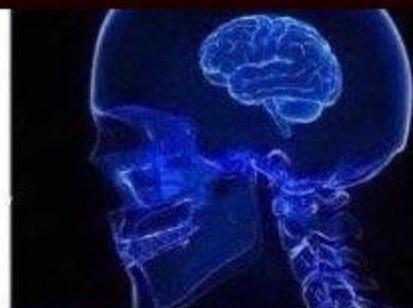


# How to download?

We wrote a python parser  
using **Beautiful Soup**  
and **Selenium** libraries



Use ready solutions



Write your own parser



Download pictures  
manually



# Filtering

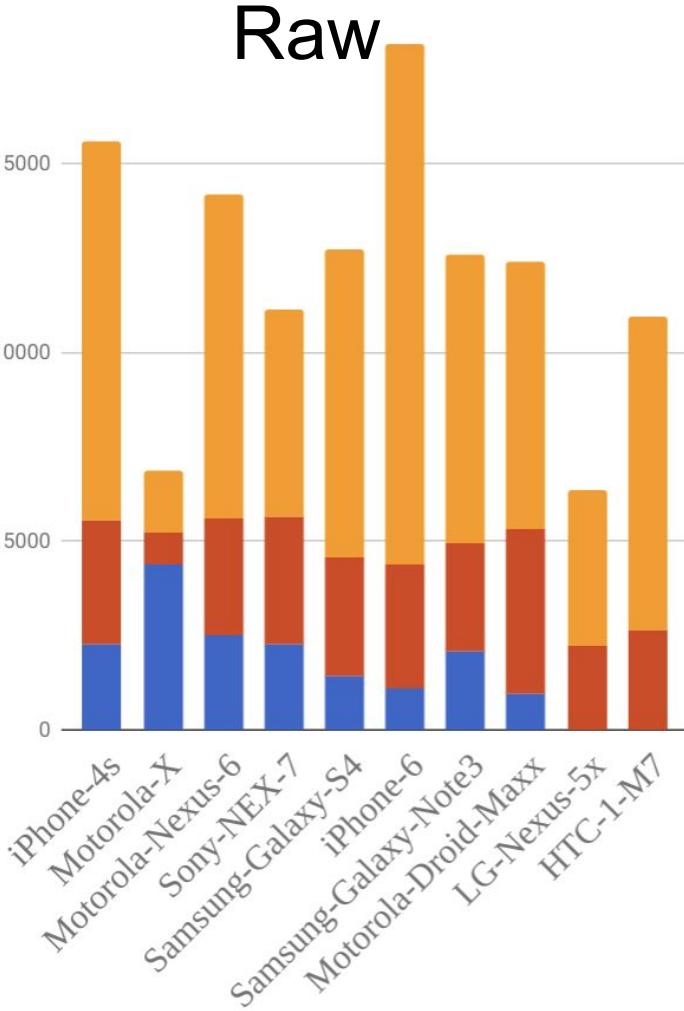
- Model
- Size
- JPG Quality
- Metadata Tags
- Processing Software



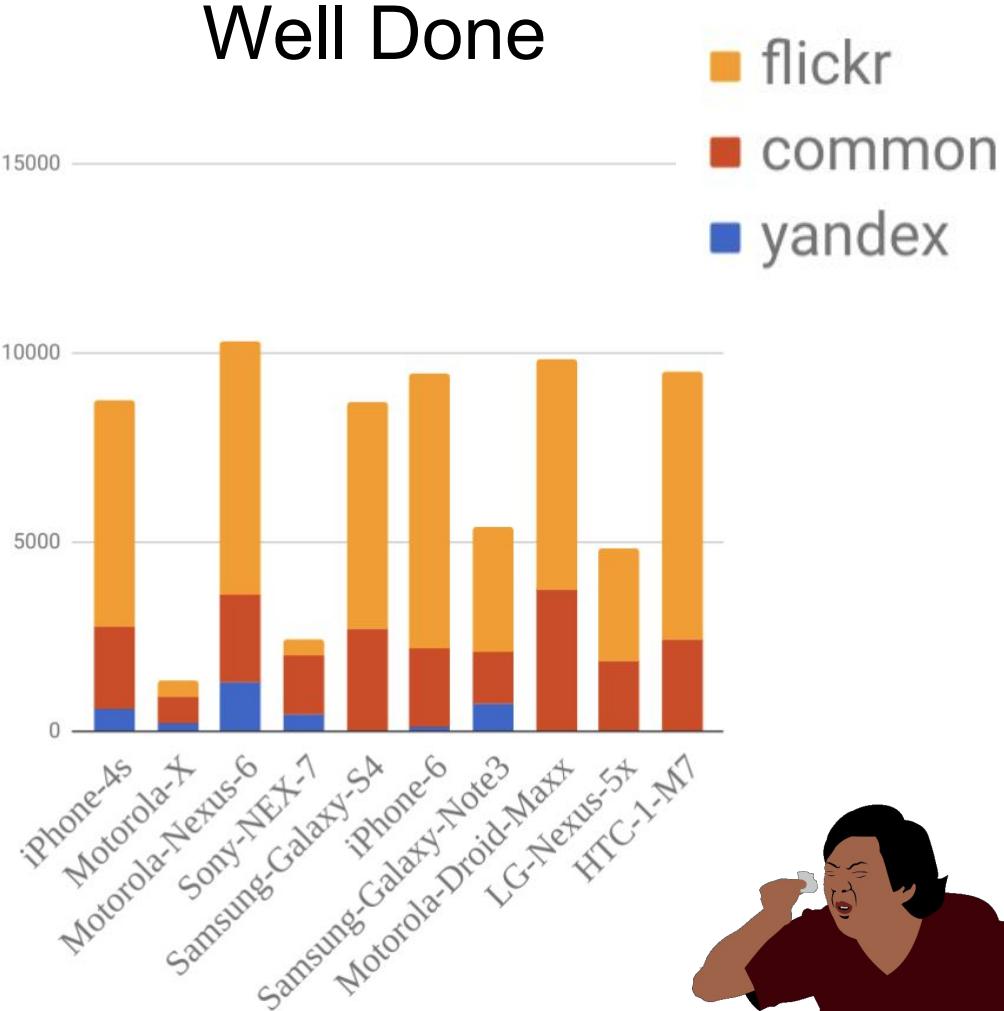
```
{  
  'name'      : 'LG-Nexus-5x',  
  'models'    : [ 'Nexus 5X' ],  
  'software'  : [ 'bullhead*' , 'HDR+*' ],  
  'shapes'    : [(4032, 3024), (3024, 4032)]  
}
```

```
{  
  'name'      : 'HTC-1-M7',  
  'models'    : [ 'HTC One' ],  
  'software'  : [ 'hvvj*' , 'HDR+*' ],  
  'shapes'    : [(2688, 1520), (2459, 1390)]  
}
```

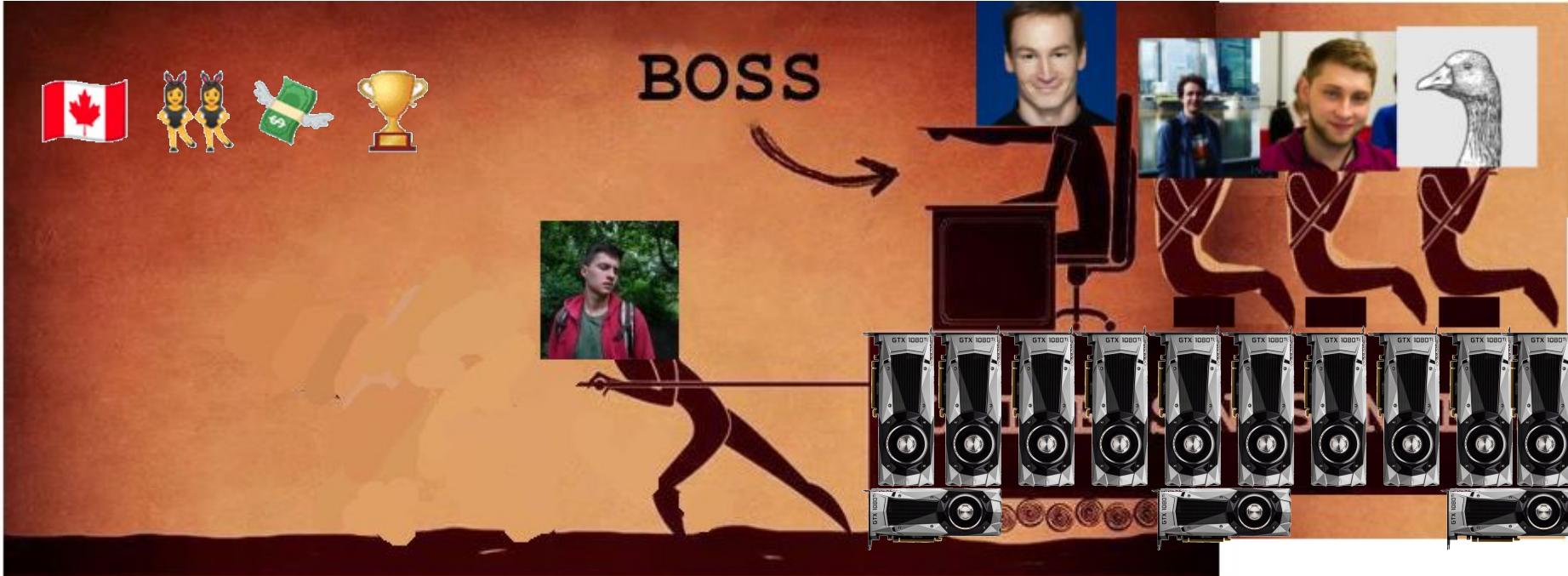
20000



20000



# Team interaction



# Credits to Andres Torrubia



Andres Torrubia

30th place

## Sharing my code, looking for suggestions (0.934 LB)

posted in IEEE's Signal Processing Society - Camera Model Identification a month ago



56

Hi guys,

I just started yesterday and I'd like to share my progress:

<https://github.com/antorsae/sp-society-camera-model-identification>



Ivan Romanov

45th place

## pytorch baseline 0.946 acc (Andres generator)

posted in IEEE's Signal Processing Society - Camera Model Identification a month ago



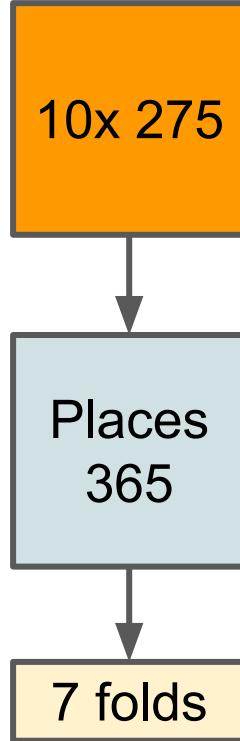
44

The code is based on Andres solution with changes made to improve performance and readability. With the original approach used as is I've managed to complete epoch in ~17 min with batch size 8 on GTX1080. After the rewrite the training time was reduced to ~4.5 while being able to fit larger batch of size 16 inside GPU memory.

<https://github.com/irrmnv/pytorch-ieee-cmi>



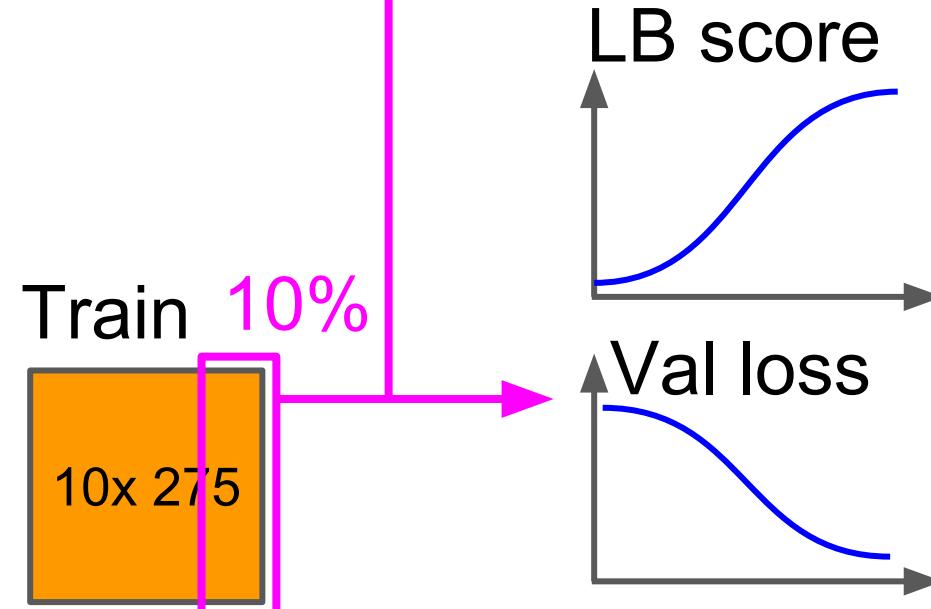
# Validation Train



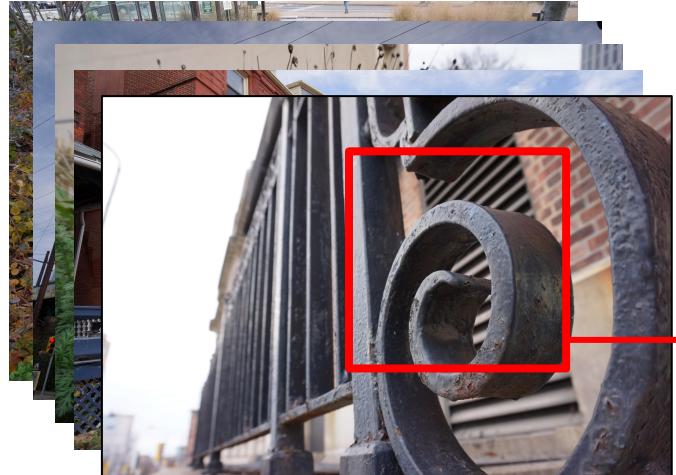
Gleb Posobin posted in IEEE's Signal Processing Society - Camera Model Identification 2 months ago

I have scrapped flickr for pictures taken with the phones from the dataset, their urls are in `flickr_images.tar.gz`. There are ten folders, in each folder there is a `urls` file (contains links to flickr pages with images) and `urls_final` (contains direct links for downloadin iaa's). all photos are

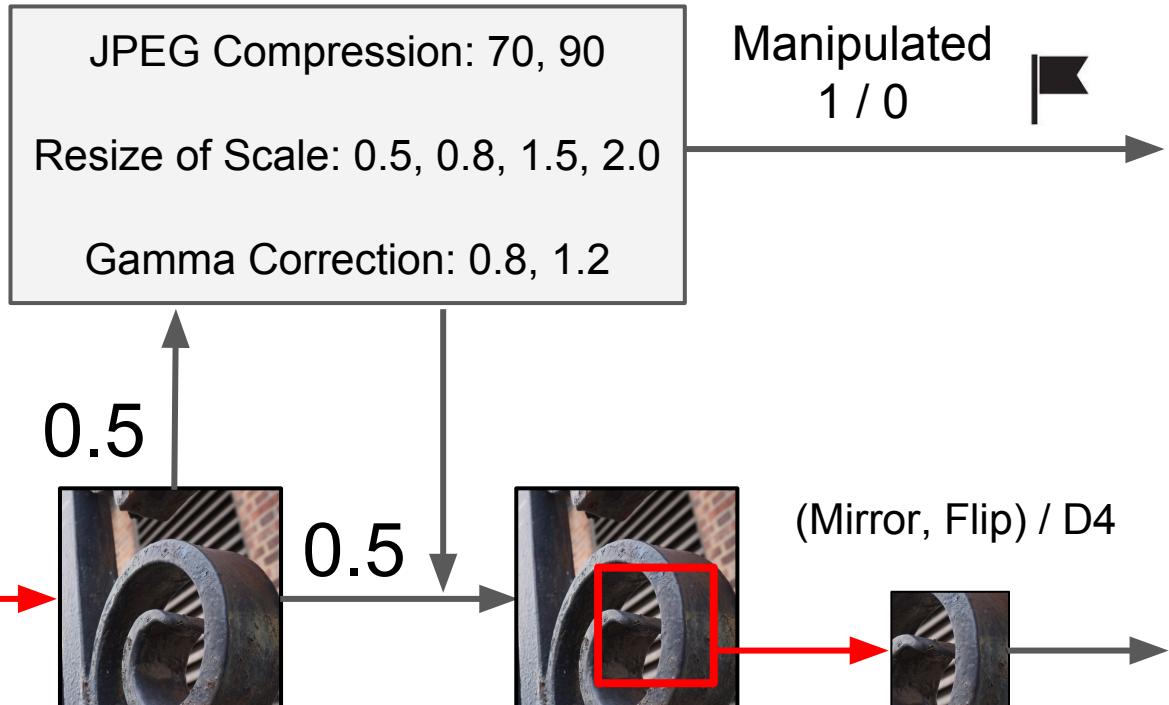
`val_images.tar.gz (3.54 KB)`  
`flickr_images.tar.gz (313.55 KB)`



# Preprocessing and Augmentation



Full Resolution Images  
Balanced Sampling



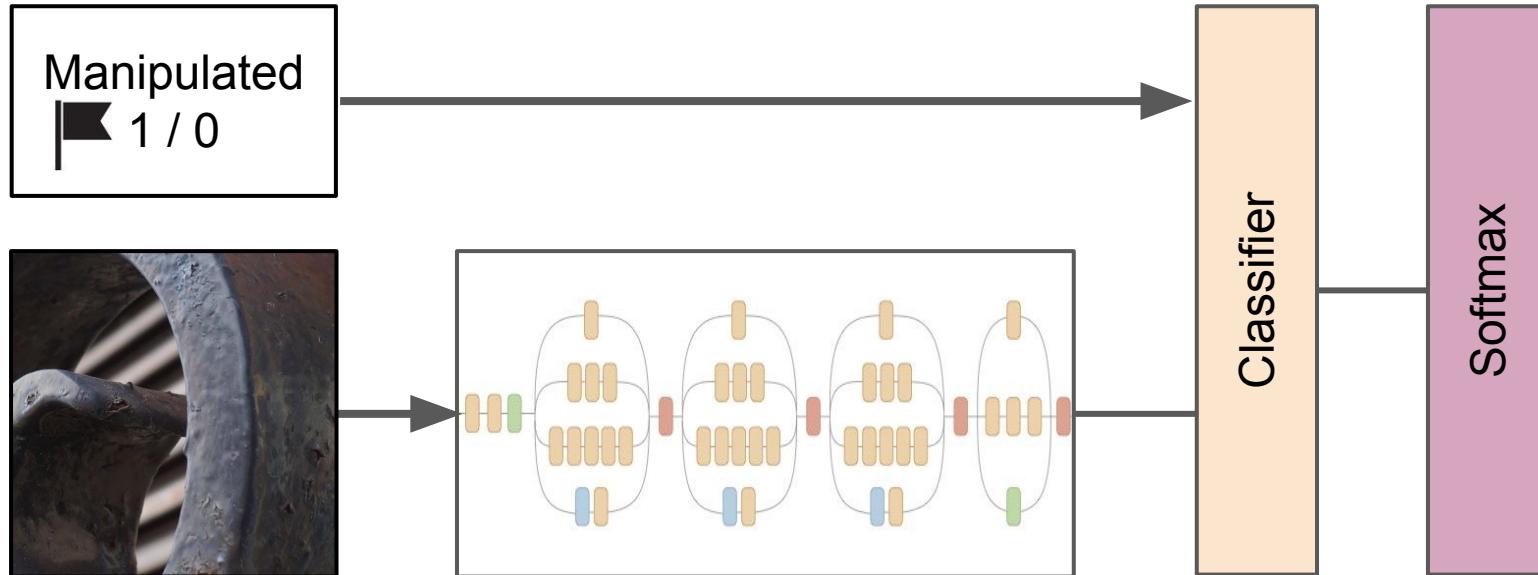
Random Crop  
960 x 960

Manipulated  
1 / 0

(Mirror, Flip) / D4

Random Crop  
480 x 480

# CNN



480 x 480,  
Random Crops  
(Mirror, Flip) / D4

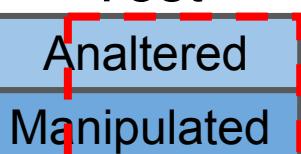
Body from:  
DenseNet-161, DenseNet-201  
ResNeXt-101  
SE-ResNet-50, SE-ResNeXt-50  
DPN-92, DPN-98

# Training

1.



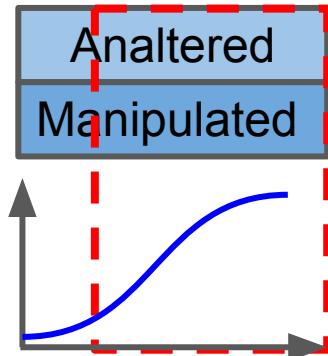
Test



80%

2.

Probs



3.



# Training Scheme

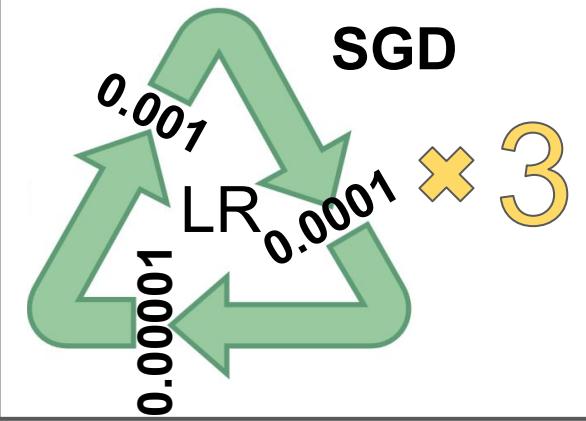


Adam

LR = 0.0001

Reduce LR on Plateau

SGD



# Test Time Augmentation

Test Image



512 x 512

TTA10 / TTA40



480 x 480

Five Crop + Orientation Flip / D4

{ }\_unalt.tif  
{ }\_manip.tif

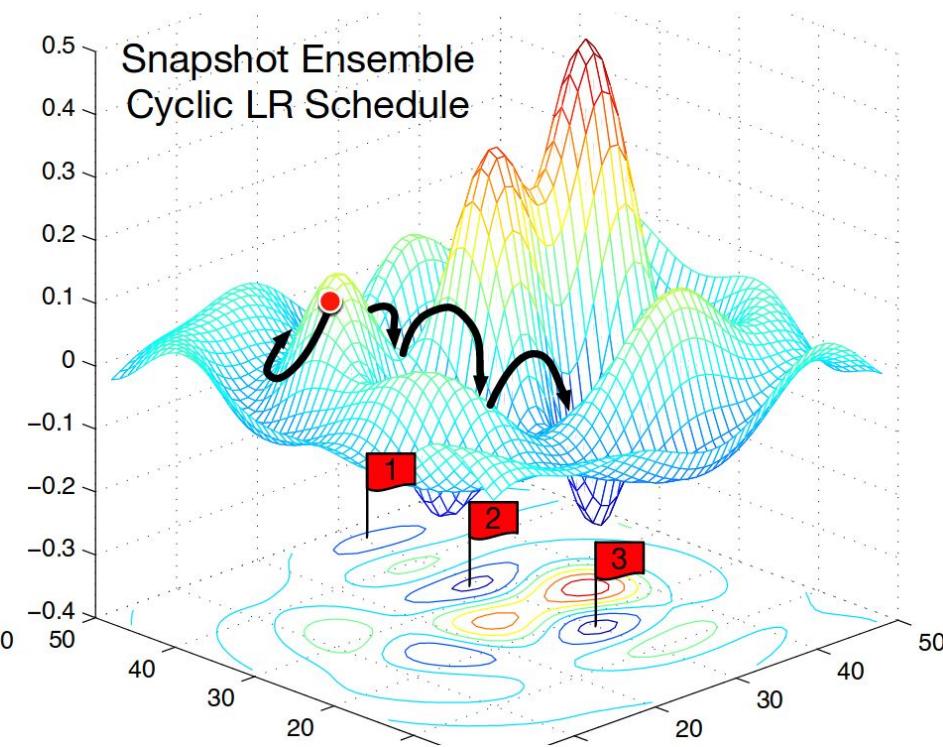
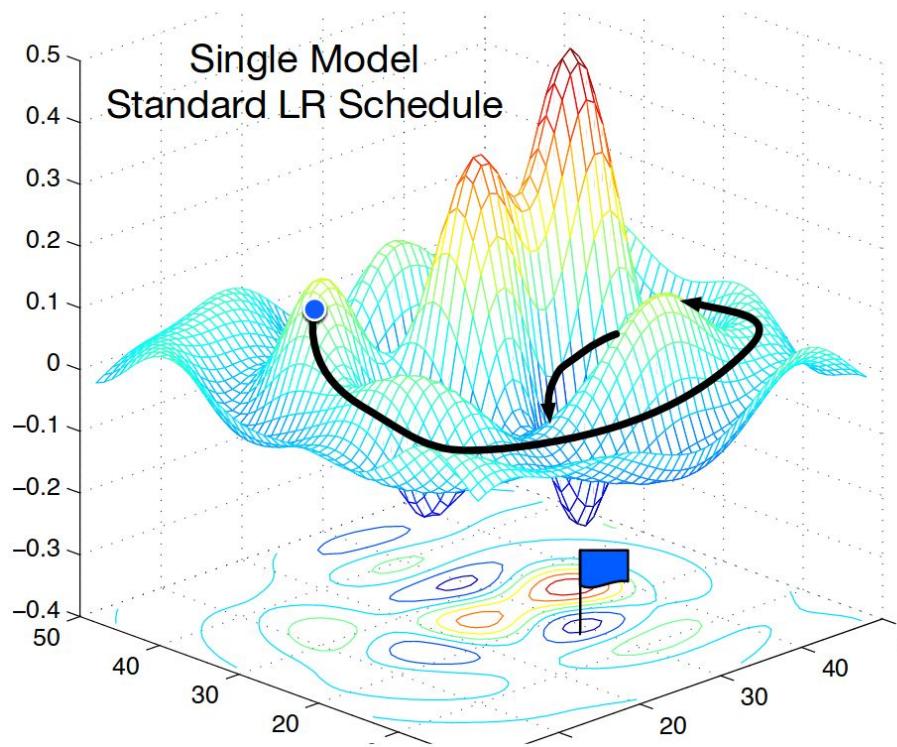
Manipulated  
1 / 0



CNN

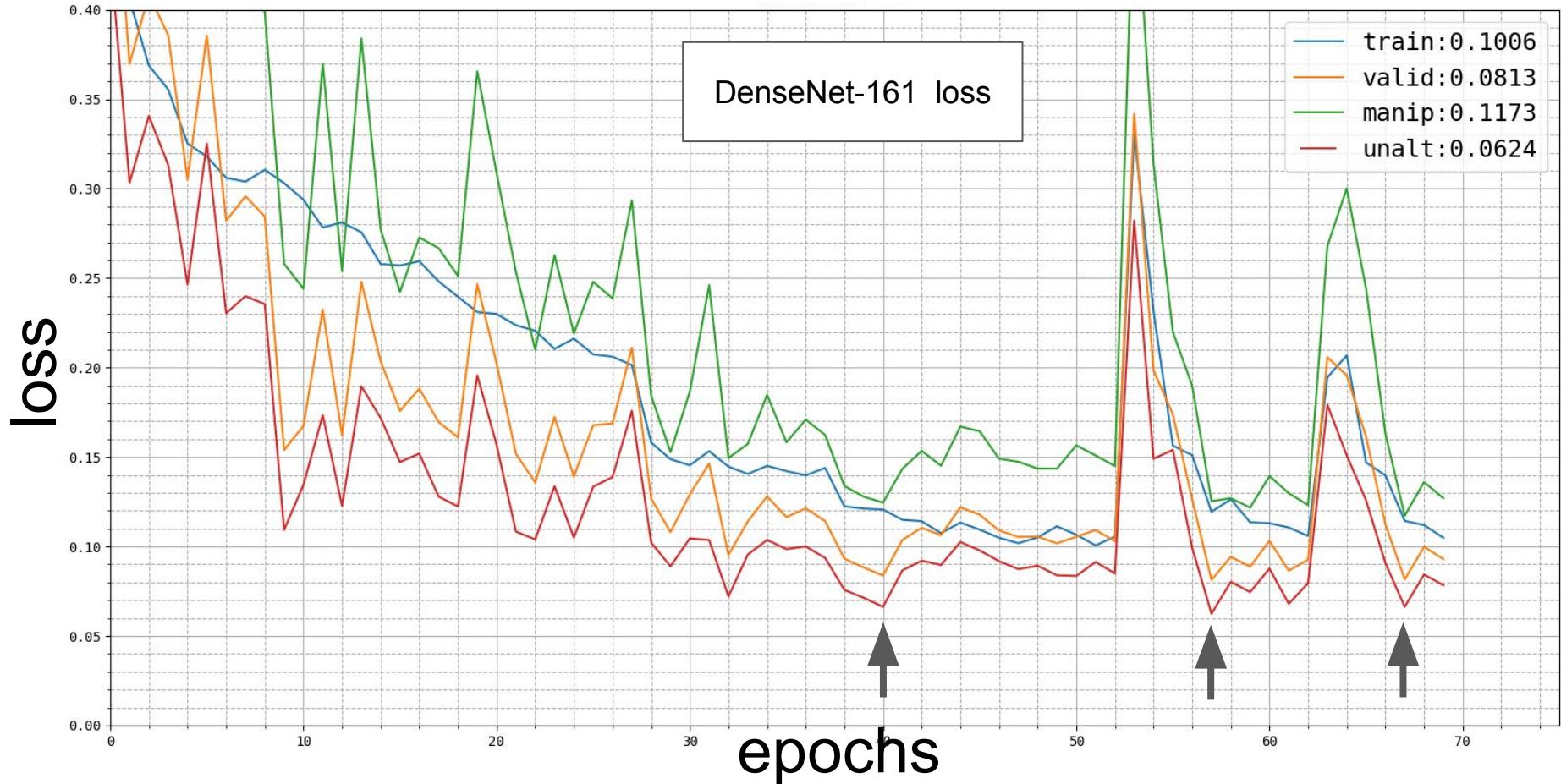
Geometric Mean

# Snapshot Ensembles (aka Multi Checkpoint TTA)

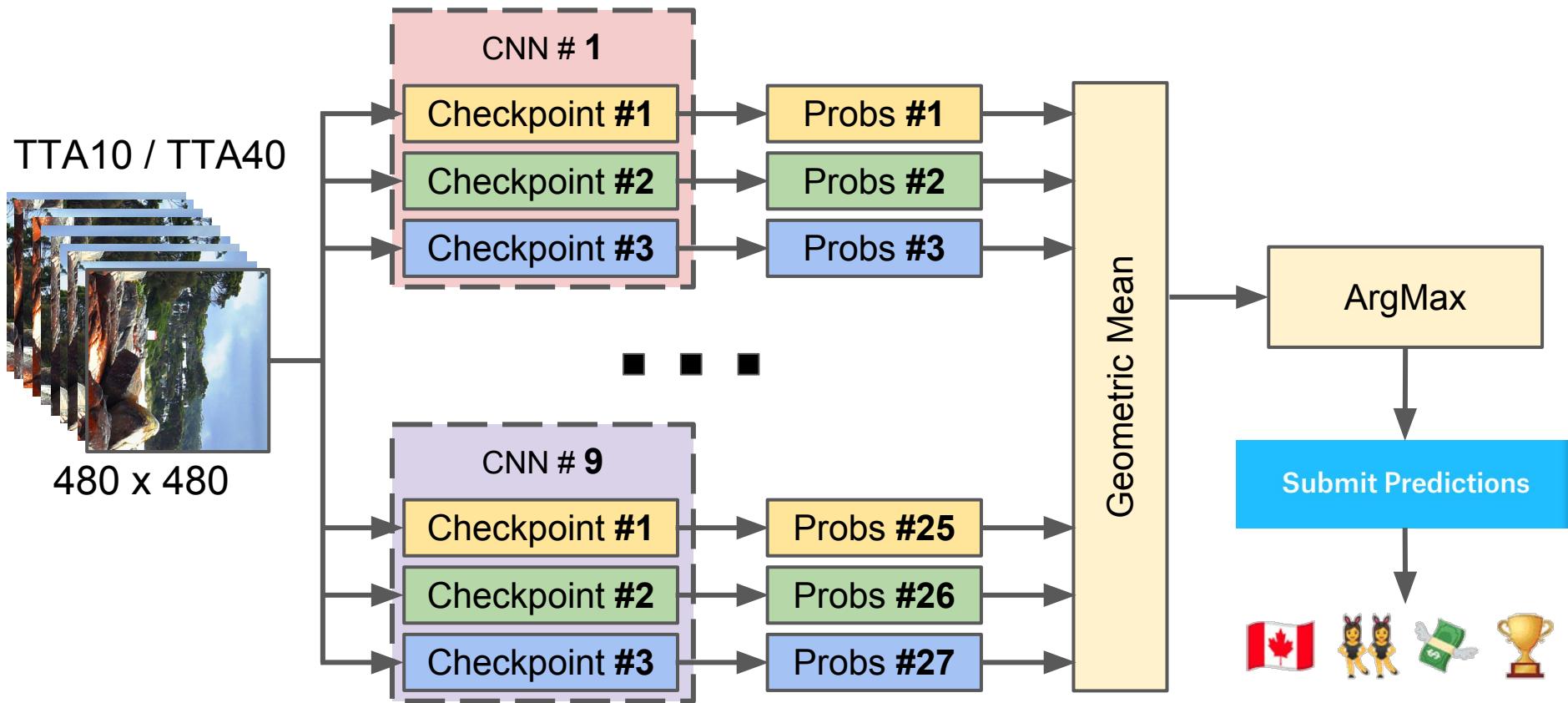


Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger.  
Snapshot Ensembles: Train 1, get M for free. <https://arxiv.org/abs/1704.00109>

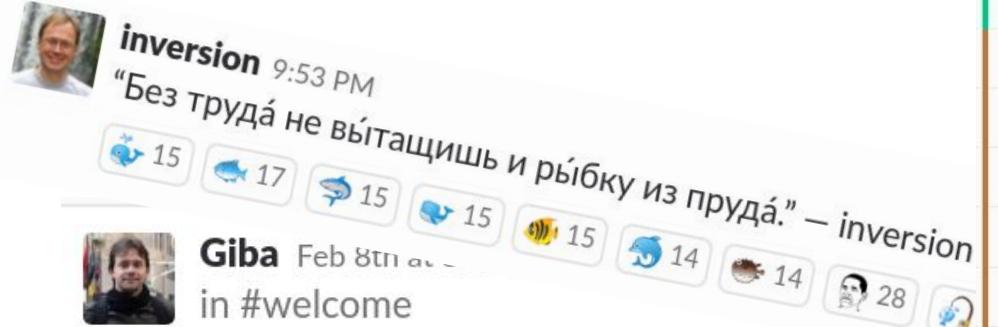
# Snapshot Ensembles



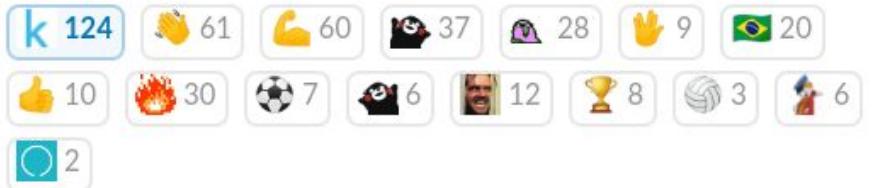
# Final Ensemble of 9 CNNs



## Results



Hello all!!!!



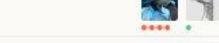
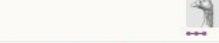
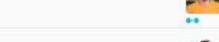
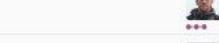
20 replies



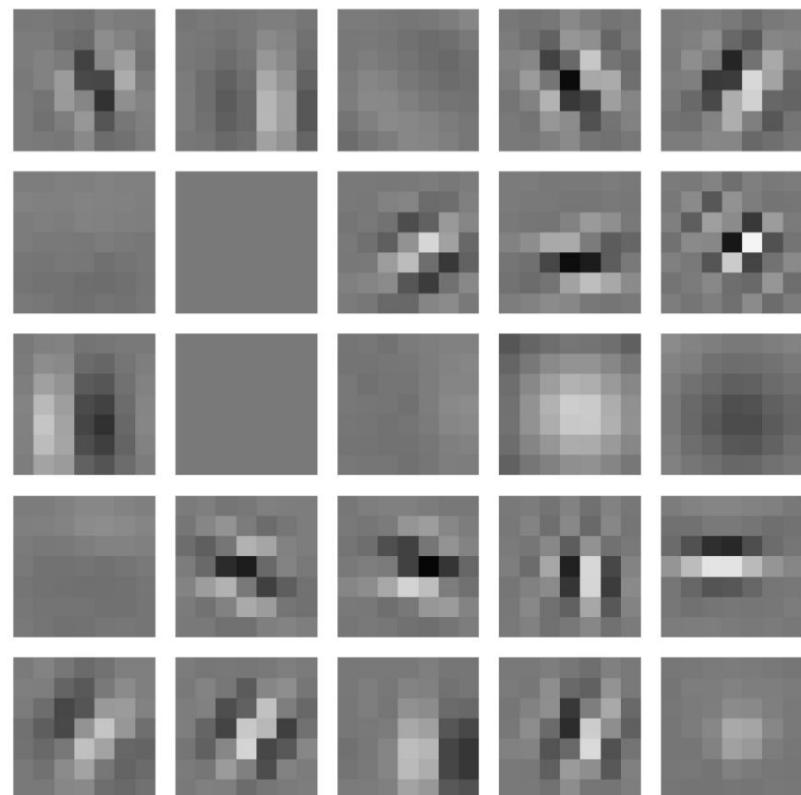
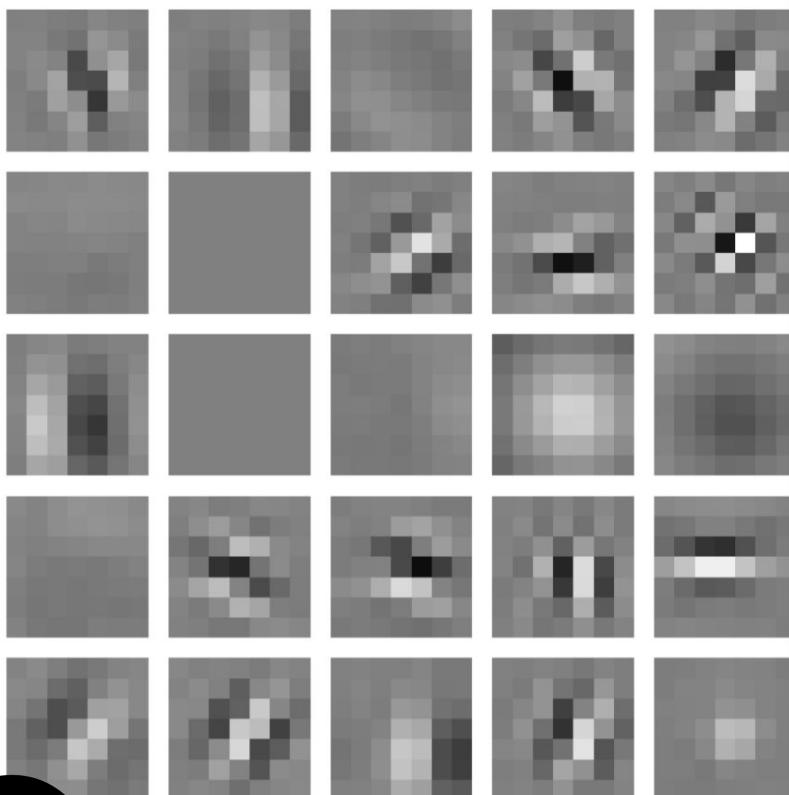
**ternaus** 18 days ago

Hi. Do you speak Russian?



1	▲ 1	[ods.ai] STAMP			0.989
2	▲ 6	[ods.ai] GPU_muscles_SPcup...			0.987
3	▼ 2	FlIGO_SPcup_eligible			0.987
4	▲ 5	Guanshuo Xu			0.987
5	▲ 2	[ods.ai] 10011000			0.986
6	▼ 3	[ods.ai] Evgeny Nizhibitsky			0.986
7	▲ 4	blzr_SPcup_eligible			0.985
8	▲ 4	Master			0.985
9	▼ 5	[ods.ai] SVM punks			0.985
10	▼ 4	Make Ensemble Great Again!			0.984
11	▼ 1	[ods.ai] Nokia3310			0.984
12	▲ 3	Yusaku   Branden   Kazanova			0.982
13	▲ 4	Bit Boys			0.982
14	▲ 7	HOHOYAO			0.981
15	▲ 10	[ods.ai] Nikolay Falaleev			0.980
16	▼ 11	[ods.ai] Alex Parinov			0.980
17	▲ 1	[ods.ai] monsta			0.979
18	▲ 11	John Doge			0.979

# Which One From ImageNet?



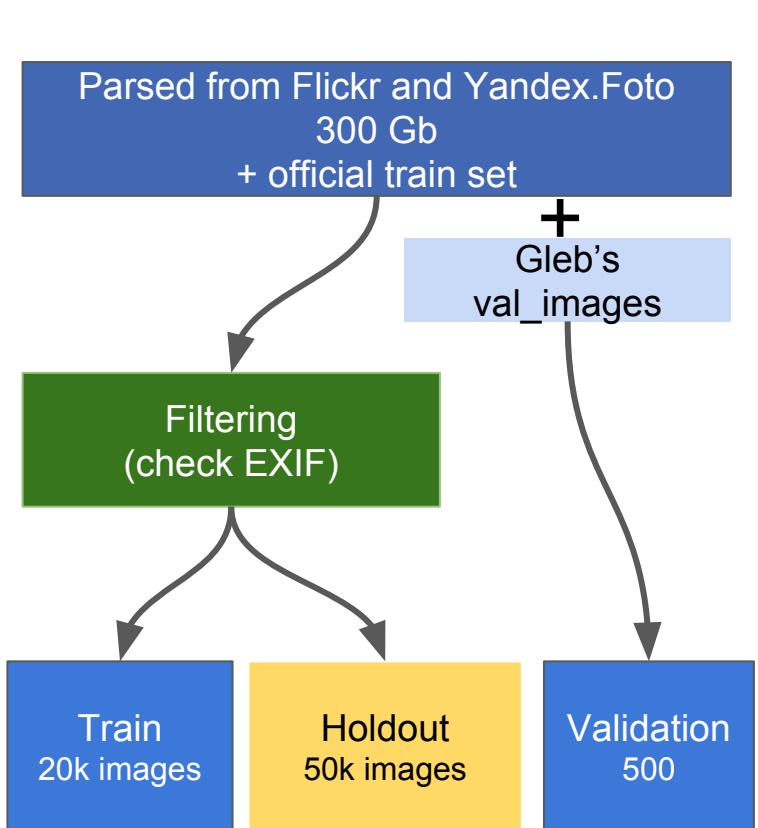
[github.com/ikibardin/kaggle-camera-model-identification](https://github.com/ikibardin/kaggle-camera-model-identification)

# IEEE's Signal Processing Society Camera Model Identification

# [ODS.ai] STAMP

Dmitriy Abulkhanov @monty\_ponty  
Sergey Mushinskiy @cepera\_ang  
Pavel Pleskov @ppleskov  
Alex Shvets @shvetsiya  
Roman Soloviev @zfturbo

# Our approach



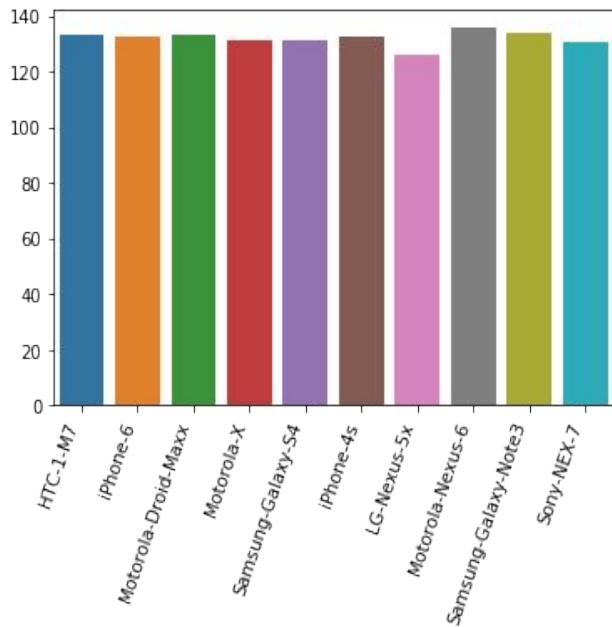
- 3x DenseNet 201 + 1x Resnet 50 + 1x InceptionResnetV2 + 1x Xception
- Train on random crops and crops from central 1024x1024 area
- Use manip flag in cnn classifier head
- Predict with D4 TTA8
- Averaging predictions by root square mean ("TTA"-wise averaging)  
+ prediction equalization  
**or**
- Put all probs into level-2 blender

# Some tricks

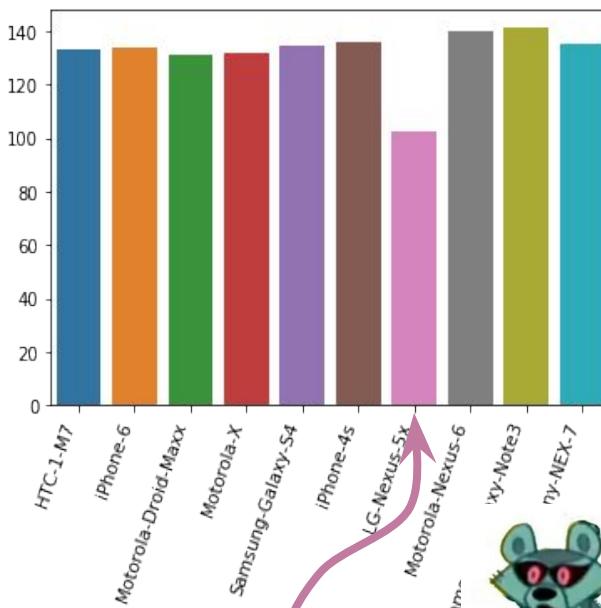
- Some models was trained on 256x256 crops and then finetuned on 512x512 (nearly 4x speed up without loss in quality )
- Cosine LR annealing on plateau
- Use “code starter kit”: half - Andres repo, half - PyTorch implementation.
- Use Selenium for parsing Flickr and Yandex.Foto
- Also trying SE-Resnet 50, MobileNet, DenseNet161

# Test set predictions imbalance

unaltered



manipulated



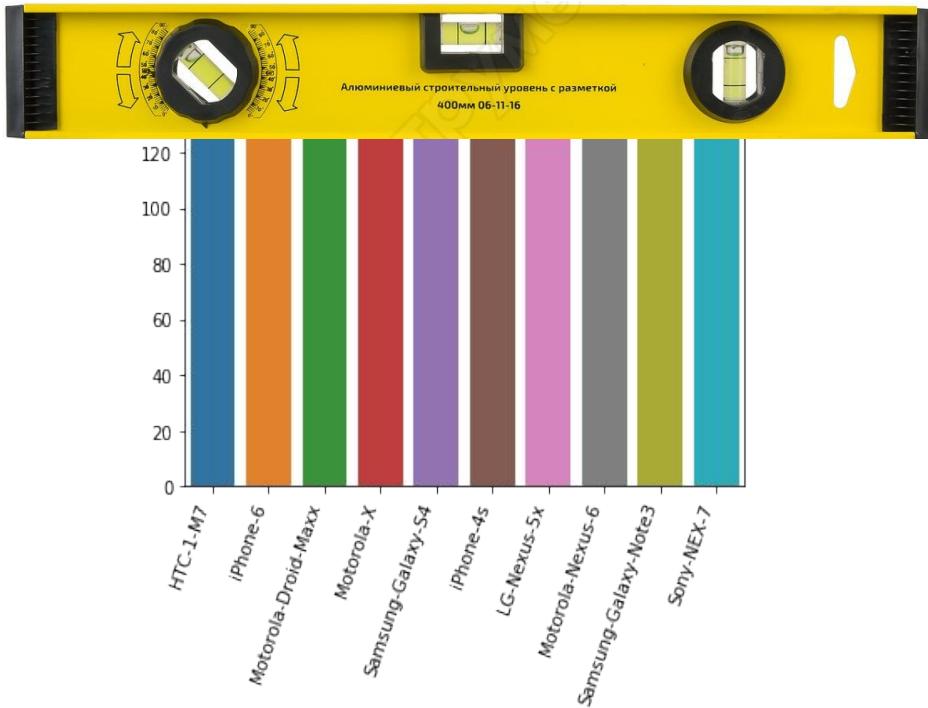
X



bicubic x0.5

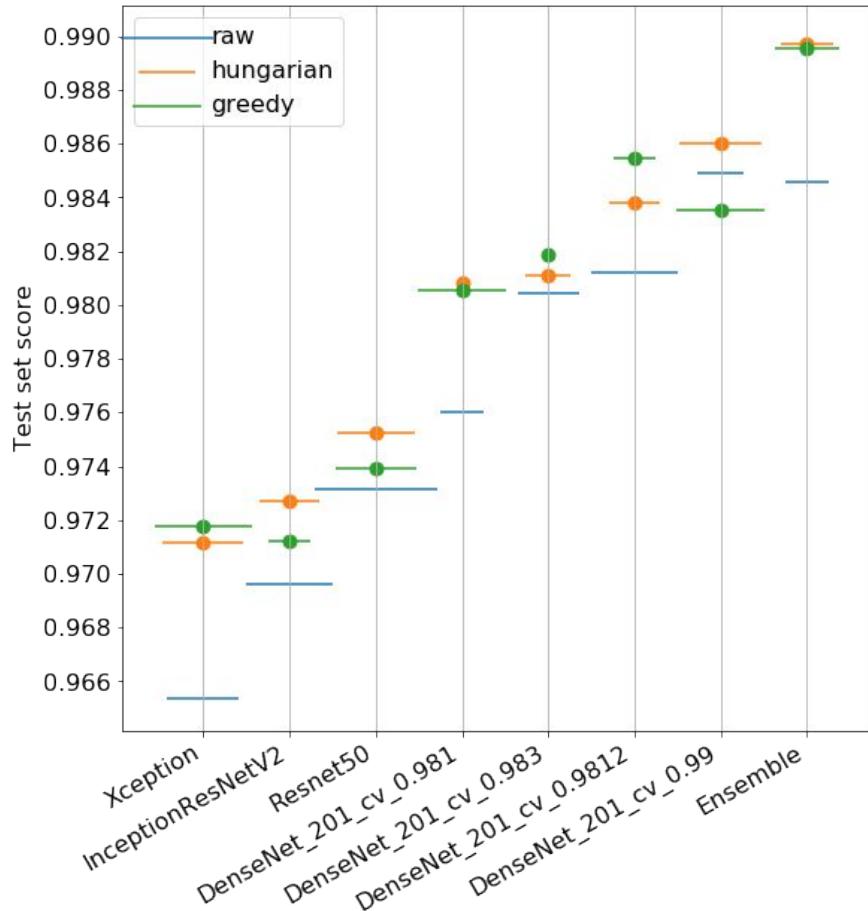
LG nexus 5x

# Prediction equalization



- Greedy algo - leave top-132 most confident samples per class (separately for unalt and manip).
- Hungarian algo - find perfect matching for complete bipartite graph with min cost (max total probability).  
Implemented in scipy

# Prediction equalization

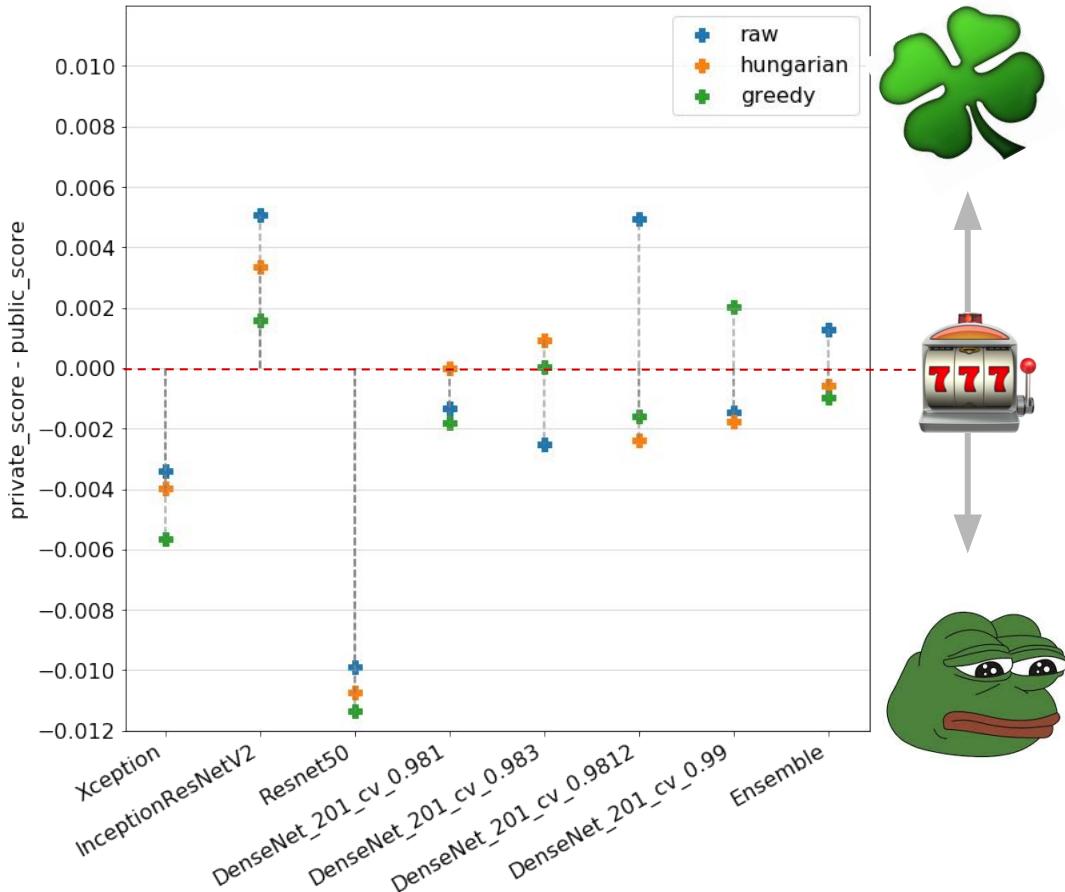


#	Score
1	0.989642
2	0.987976
3	0.987857
4	0.987023
5	0.986547
6	0.986190
7	0.985595
8	0.985595
9	0.985357
10	0.984761
11	0.984404
12	0.982738
13	0.982142

Pros:

- “unsupervised” - no need to fit level-2 model
- score increasing 0.001 - 0.006 (+1 - +10 up at LB)
- increase almost in any case

# Prediction equalization



Cons:

- increase score dispersion
- LB score become more random 
- need better validation  
(contradiction with prediction imbalance problem statement)

# Level-2 blender



**XGBoost (20) + LightGBM (20) + Keras (12)**

- Increase class\_weight for nexus 5x, decrease for motorola nexus 6
- train 52 good level-2 models on huge holdout (50k images)
- blend'em all!
- give a slightly lower public LB score than the ensemble with prediction equalization, but is more robust.

# Things, which did not work/used

- Dependency between manipulation and file size (RLE compression feature, pic. 1)
- Time structure in test set dates of creation (pic. 2)
- Extract noise from image and concatenate with input channels
- Different design of classifier head in cnn's

