



MobileODT  
Mobile Optical Detection Technologies

## Определение типа шейки матки по изображениям (Kaggle Intel & MobileODT Cervical Cancer Screening)



Байкулов Руслан  
[kaggle.com/romul0212](https://kaggle.com/romul0212)  
ODS: @romul

# Постановка задачи

По изображению нужно определить тип матки.

Разные типы матки -  
разные зоны эрозии.

Определение типа критично  
важно для выбора  
подходящего метода  
лечения рака шейки матки.

Метрика качества logloss

## Type 1

- Completely ectocervical
- Fully visible
- Small or large



## Type 2

- Has endocervical component
- Fully visible
- May have ectocervical component which may be small or large



## Type 3

- Has endocervical component
- Is not fully visible
- May have ectocervical component which may be small or large

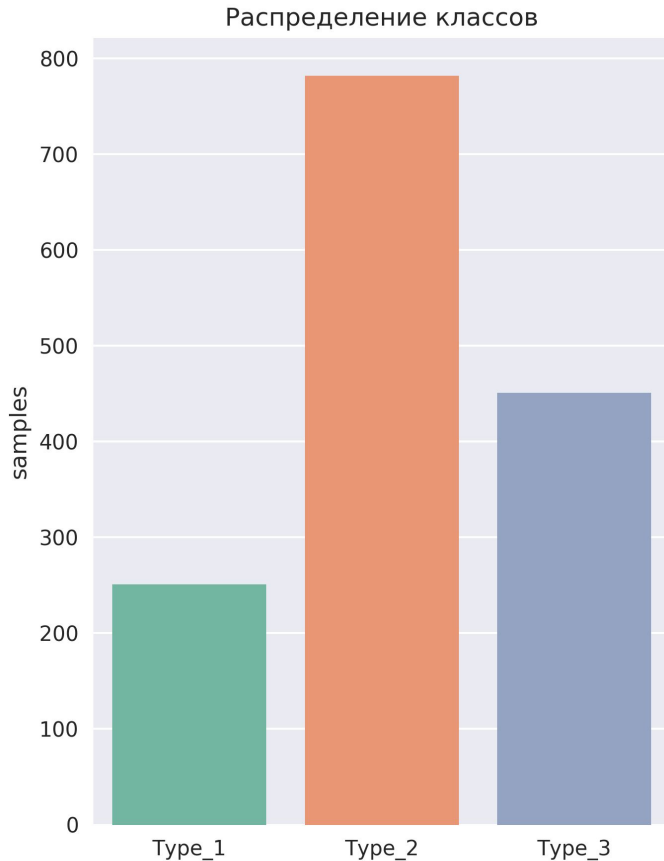


# Данные

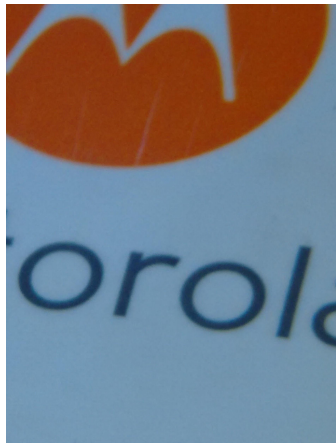
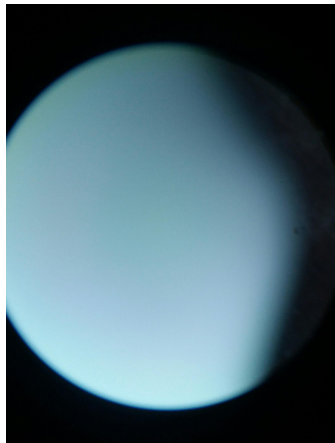
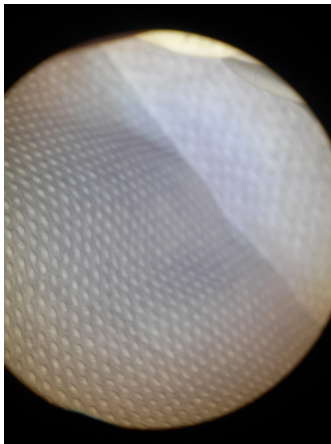
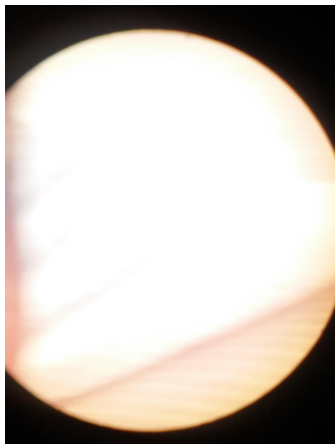
- Train - 1481 images (6.1 GB)
- Additional - 6734 images (28.8 GB)
- Test stage 1 - 512 images (2.1 GB)
- Test stage 2 - 3506 images (9.4 GB)

В additional много проблем:

- Неправильная разметка
- Пересечение с train и **test** выборкой
- Фотографии сделанные с помощью ПНВ (зеленые)
- Размытые фотографии
- Фотографии не относящиеся к задаче

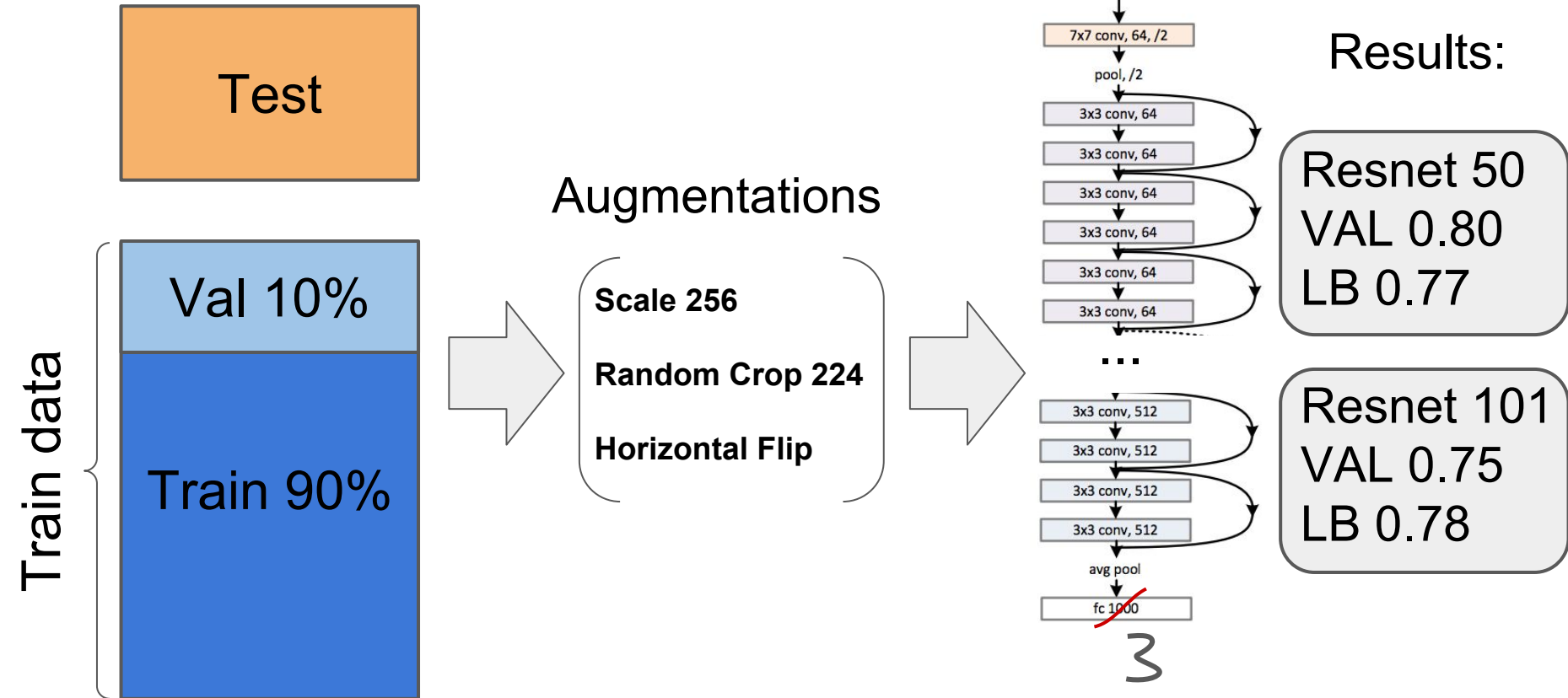


# Примеры изображений из additional



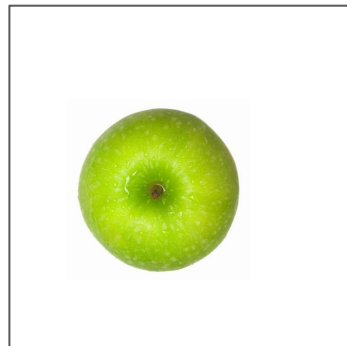
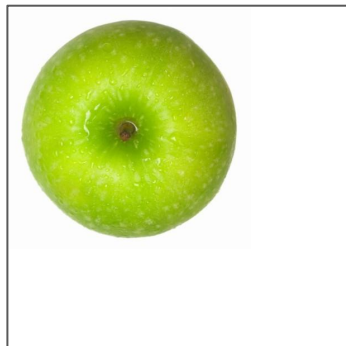
# Первые попытки

## Resnet



# Разный масштаб

При изменении размера изображения можно потерять много информации.



Хочется научиться находить полезную часть снимка и обучать модель только на ней.

# Разный масштаб

При ресайзе изображения можно потерять много информации.

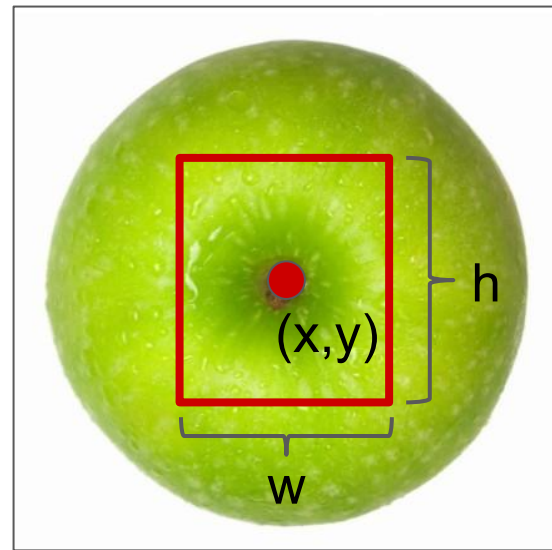
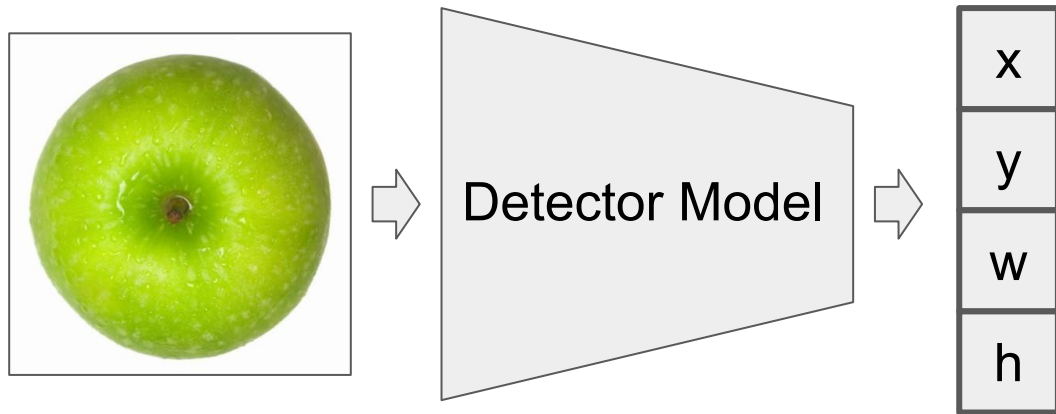


Хочется научиться находить полезную часть кадра и обучать модель только на ней.



# Детектор

Обучаем нейронную сеть предсказывать координаты центра, ширину и высоту прямоугольника.



37



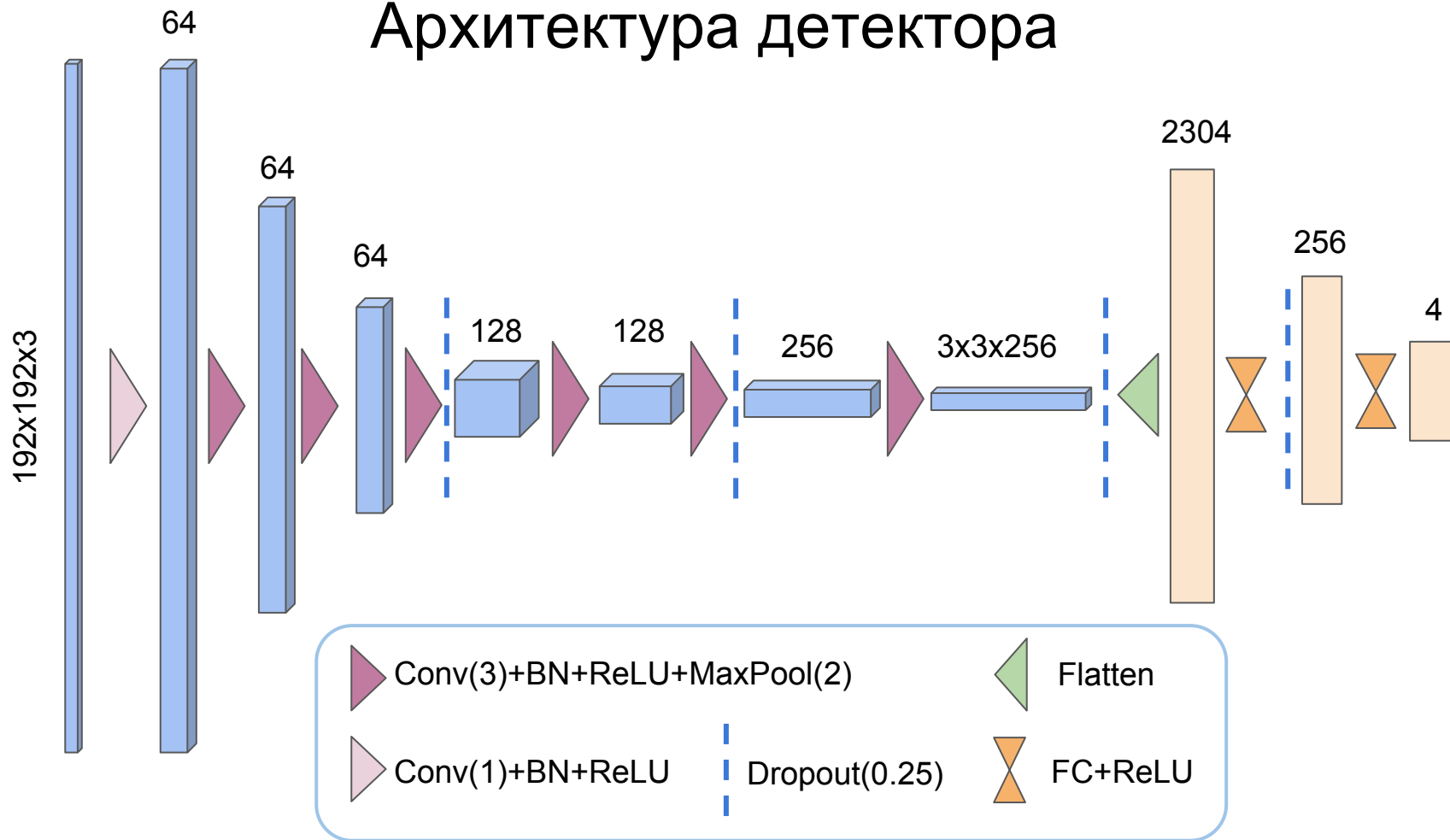
**Bounding boxes for Type\_1**

posted 3 months ago by Paul

<https://www.kaggle.com/deveaup>

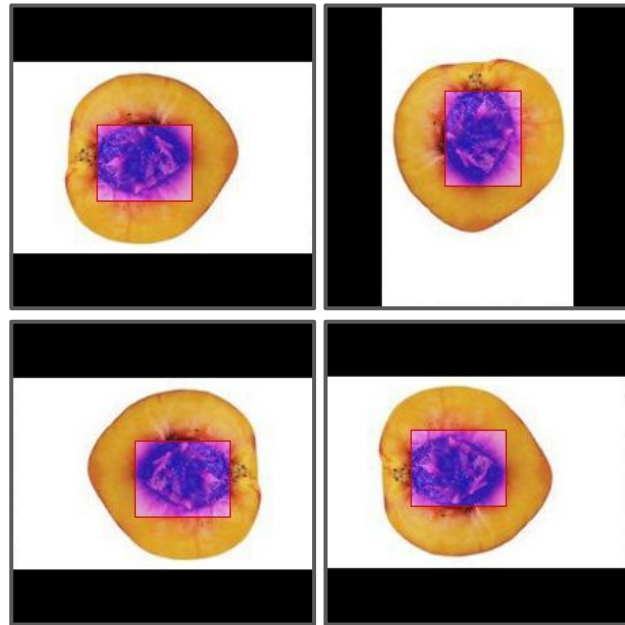
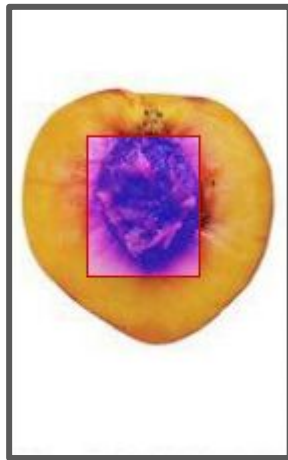


# Архитектура детектора



# Предобработка и аугментация изображений для детектора

- Resize with zero padding
- Rotate 90, 180, 270 degrees
- Horizontal and vertical flips



# Обучение детектора

- Loss: MAE
- Optimizer: Adam
- Batch size: 32
- Learning rate: 0.001
- Number of epoch: 200

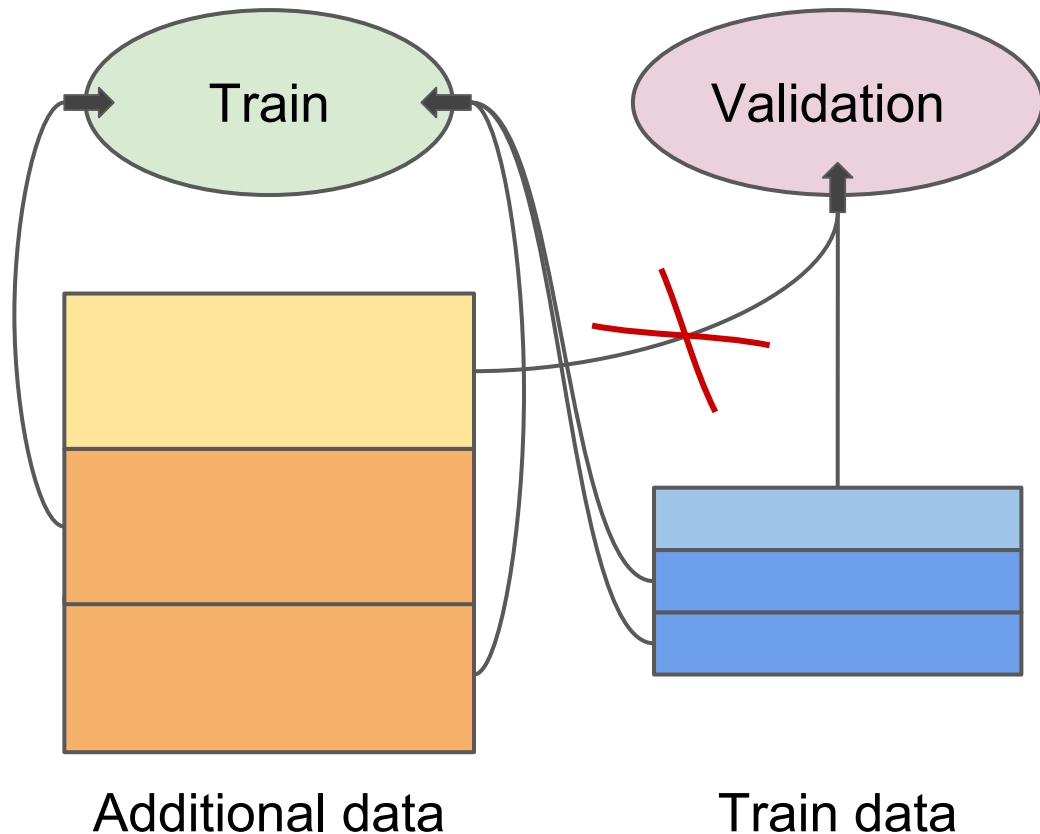
```
Epoch 0: train 17.053188624920875    val 13.231120122445596
Epoch 1: train 13.91152842263423      val 12.903140235591579
Epoch 2: train 13.414280067883066     val 12.31360117164818
Epoch 3: train 13.02453057954052     val 12.816465699994886
Epoch 4: train 12.797467422892035     val 11.929914165187526
Epoch 5: train 12.32286236687764      val 11.604760234420365
Epoch 6: train 12.161445806783908     val 11.728974097483867
Epoch 7: train 11.87443016027845      val 11.334061274657378
Epoch 8: train 11.57122896068386      val 11.51347659085248
Epoch 9: train 11.405853763572189     val 10.550211932208088
Epoch 10: train 11.232540472229914    val 11.491866910779798
Epoch 11: train 10.983012770793078    val 10.162859613830978
Epoch 12: train 10.772776530495584    val 10.252024019086683
Epoch 13: train 10.648172998733358    val 9.847372274141055
```

■ ■ ■

```
Epoch 195: train 5.466486436725933   val 9.470098289283547
Epoch 196: train 5.492792625671257   val 9.43220431095845
Epoch 197: train 5.480475711415826   val 9.580626023782266
Epoch 198: train 5.439806982906642   val 9.404440538303271
Epoch 199: train 5.445368868455704   val 9.484641216896676
Epoch 200: train 5.419769179338077   val 9.644662676630793
Epoch 201: train 5.46945775749841    val 9.311708392323675
```

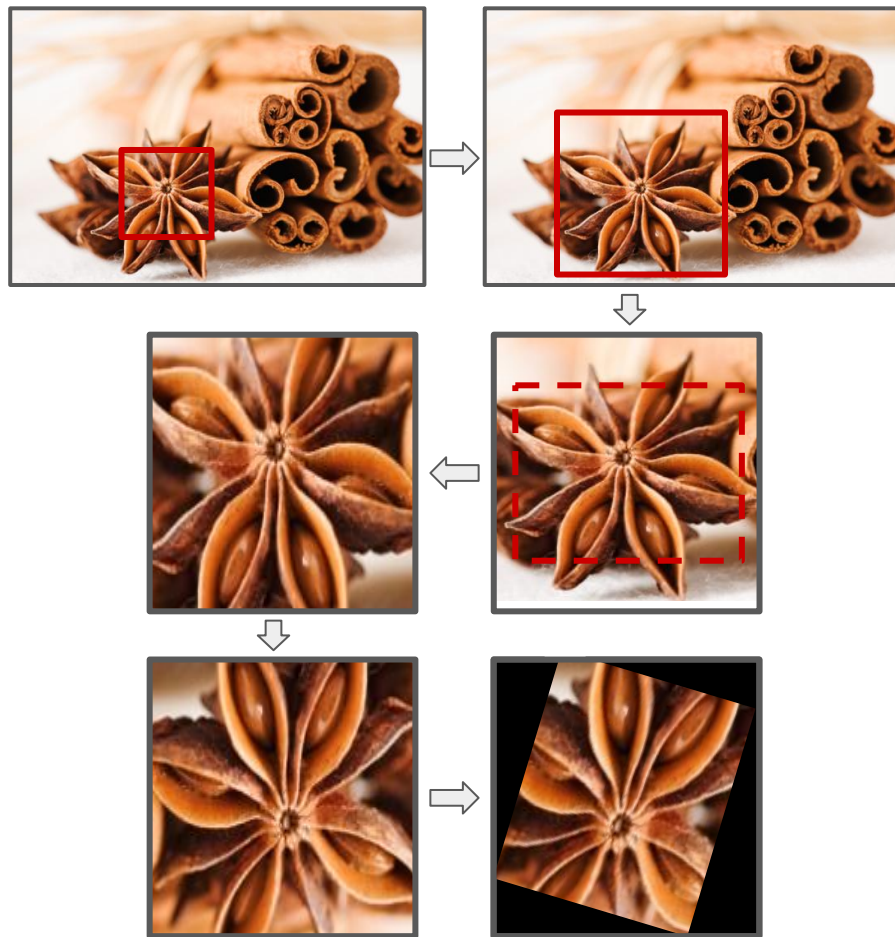
# Валидация с additional

- Train 10 folds
- Additional 10 folds
- Не использовать additional данные при валидации моделей
- Обучающая выборка =  $4 * \text{train} + \text{additional}$



# Предобработка и аугментация изображений для классификатора

- Crop rectangle from detection model
  - scale width and height by N times
- Resize image to 256x256
- Random sized crop
  - random size crop of (0.5 to 1.0) of the original size and a random aspect ratio of 3/4 to 4/3
  - Resize to 224x224
- Horizontal and vertical flips
- Random rotate 360 degrees



# Финальная модель

Два усреднения 10-фолдного файнтюна VGG16.

Отличаются тем, во сколько раз увеличивается ширина и высота прямоугольников из детектора (в 1.5 и 2 раза).

Параметры при обучении моделей:

- Loss: logloss
- Optimizer: SGD  
Momentum: 0.9, Weight decay: 1e-4
- Batch size: 16
- Learning rate: 0.0001
- Number of epoch: 25

Final model	
Val	0.65720
Public	0.61245
Private	0.81809

# Технические подробности

- PyTorch
- Docker
- Видеокарта: Nvidia GTX 1060 6Gb
- Время обучения всех моделей примерно сутки
- Github: [github.com/IRomul/intel\\_cancer](https://github.com/IRomul/intel_cancer)



# Факап с предсказаниями

За полтора часа до окончания соревнования скачал финальный сабмит и заметил это!



















Все изображения из stage 2 не были предсказаны.

После исправления ошибки примерно за час удалось получить предсказания со всех 20 моделей.


```
504 502.jpg,0.05126477978192269,0.31789873316884043,0.6308364942669868
505 503.jpg,0.03148906810674816,0.4965880401432514,0.4719228900969029
506 504.jpg,0.0144525762880221,0.1675898505374789,0.8179575681686402
507 505.jpg,0.02493442359846085,0.9444483637809752,0.030617220074054787
508 506.jpg,0.20690540559589862,0.5856685861945152,0.2074260089546442
509 507.jpg,0.28801270648837085,0.5969220012426376,0.11506527578458192
510 508.jpg,0.06714690998196603,0.7199251025915145,0.21292797983624043
511 509.jpg,0.014323029638035223,0.8790652036666871,0.10661174384877087
512 510.jpg,0.01779857851797715,0.3956246078014374,0.5865768179297447
513 511.jpg,0.028614795906469228,0.22943555265665055,0.7419496566057205
514 10000.jpg,0.16999999999999998,0.5300000000000001,0.29999999999999993
515 10001.jpg,0.16999999999999998,0.5300000000000001,0.29999999999999993
516 10002.jpg,0.16999999999999998,0.5300000000000001,0.29999999999999993
517 10003.jpg,0.16999999999999998,0.5300000000000001,0.29999999999999993
518 10004.jpg,0.16999999999999998,0.5300000000000001,0.29999999999999993
519 10005.jpg,0.16999999999999998,0.5300000000000001,0.29999999999999993
520 10006.jpg,0.16999999999999998,0.5300000000000001,0.29999999999999993
521 10007.jpg,0.16999999999999998,0.5300000000000001,0.29999999999999993
522 10008.jpg,0.16999999999999998,0.5300000000000001,0.29999999999999993
523 10009.jpg,0.16999999999999998,0.5300000000000001,0.29999999999999993
524 10010.jpg,0.16999999999999998,0.5300000000000001,0.29999999999999993
525 10011.jpg,0.16999999999999998,0.5300000000000001,0.29999999999999993
526 10012.jpg,0.16999999999999998,0.5300000000000001,0.29999999999999993
```

```
# SAMPLE_PATH = '/workdir/data/sample_submission.csv'
SAMPLE_PATH = '/workdir/data/sample_submission_stg2.csv'
```

# Private Leaderboard

#	△pub	Team Name	Kernel	Team Members	Score ?	Entries	Last
1	▲125	Towards Empirically Stable Trai...		  	0.76964	2	10d
2	▲114	I Rustandi			0.80278	8	9d
3	▲79	GRXJ		   	0.80830	5	11d
4	▲94	Ruslan Baikulov			0.81809	4	9d
5	▲73	BMCi			0.82205	5	9d
6	▲80	kubilai			0.82496	26	10d
7	▲124	Shai			0.82850	2	15d
8	▲93	Ryan Munion			0.83131	3	9d
9	▲78	ZFTurbo			0.83209	2	12d
10	▲134	Alexander Popov			0.83334	19	10d
11	▲136	mokp			0.83367	2	15d
12	▲188	utility			0.83517	2	12d
13	▲129	Praveen Adepu			0.83536	3	11d

# Решение победителей ZadraraS, raddar, bobutis

#	Δpub	Team Name	Kernel	Team Members	Score ?	Entries	Last
1	▲125	Towards Empirically Stable Trai...		 ●●●●● ●●●●● ●●●●●	0.76964	2	10d

Залог успеха - правильная валидация.

Команда поняла, что данные из additional сильно пересекаются с train. Был большой датасет, в котором было по несколько фотографий от одного пациента. Для каждого пациента выбрали лучшую и поместили в train, остальное в additional.

Перед тем как бить на фолды, сделали кластеризацию:

1. calculate image colour histograms
2. run k-means clustering with  $k = 100$
3. take random 20 clusters and use that for validation

# Решение ZFTurbo



Роман Соловьев, Ph.D.  
[kaggle.com/zfturbo](https://kaggle.com/zfturbo)  
ODS: @zfturbo

9 ▲78 ZFTurbo

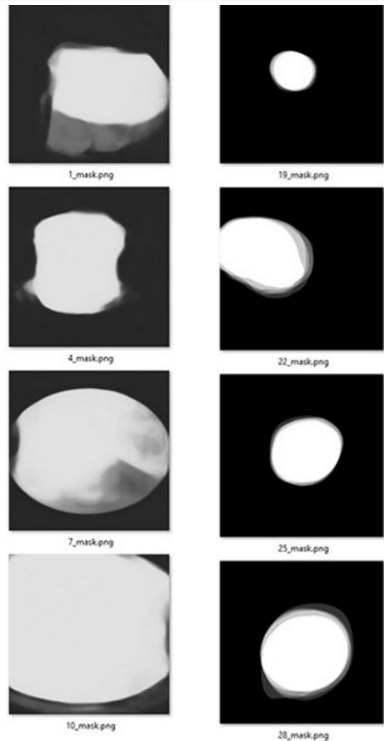


0.83209

2

12d

- Сегментация ROI с помощью UNET
- Аугментации:
  - random crops based on Unet predictions
  - random rotations
  - lightning change
  - rare random blur
- Модели: VGG16, VGG19, Resnet50, InceptionV3, SqueezeNet, Densenet161, Densenet121
- Валидация 5 Fold
- XGBoost blender
- Лучший сабмит получился без additional



Спасибо за внимание!  
Вопросы?