



Planet: Understanding the Amazon from Space

Решение от Романа Соловьева и Станислава
Семёнова (3 место в финальной таблице)

Команда

Роман Соловьев: <https://kaggle.com/zfturbo>

- Ведущий научный сотрудник ИППМ РАН
- к.т.н. в области микроэлектроники и САПР

Станислав Семёнов: <https://kaggle.com/stasq7>

- Степень магистра в Computer Science
- 2 место в рейтинге Kaggle. Более 10 побед и 28 золотых медалей.

До (Public)

#	Δ1w	Team Name	I model uploaded * in the money	Score
1	—	SYSU CISLab	*	0.93448
2	↑4	Russian Bears	‡ *	0.93448
		• ZFTurbo • Stanislav Semenov		
3	↑4	Urucu	*	0.93443
4	↑38	Plant	‡	0.93431
5	↓2	Team samogon		0.93422
6	↑11	Clear Sky		0.93419
7	↓5	Kyle		0.93412
8	↑4	ods.ai		0.93408
		• Konstantin Lopuhin • Evgeny Nizhibitsky • Alexander Buslaev • Vladimir Iglovikov • n01z3 • Ruslan Baikulov • Alexey Noskov		
9	↓4	bestfitting	‡	0.93398
10	↓6	deebsense.io		0.93392
11	↓1	Moss		0.93386

После (Private)

#	Δrank	Team Name	I model uploaded * in the money	Score
1	↓8	bestfitting	‡ *	0.93317
2	↓2	Plant	‡ *	0.93294
3	↓1	Russian Bears	‡ *	0.93277
		• ZFTurbo • Stanislav Semenov		
4	↑8	Daniel FG		0.93271
5	↓1	Clear Sky		0.93269
6	↓5	SYSU CISLab		0.93265
7	↓1	ods.ai		0.93224
		• Konstantin Lopuhin • Evgeny Nizhibitsky • Alexander Buslaev • Vladimir Iglovikov • n01z3 • Ruslan Baikulov • Alexey Noskov		
8	↓7	dhammad		0.93221
9	↓10	CCT		0.93220
10	—	deebsense.io		0.93216
11	↓4	Kyle		0.93211

Постановка задачи

Дан набор изображений частей снимков со спутника: 256 на 256 пикселей.

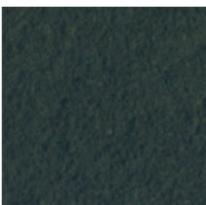
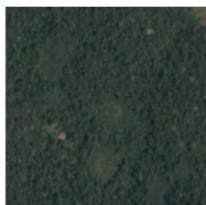
- Предоставлено 2 набора изображений JPG (3 канала) и TIFF (4 канала)
- В тренировочной части 40479 изображений
- В тестовой части 61191 изображений (~20К было добавлено после обнаружения утечки)
- Каждому изображению нужно найти набор категорий, к которым оно относится (всего 17 классов)
- Категорий у изображения может быть несколько. 4 категории относятся к погоде, 13 к поверхности

Метрика F2 score:

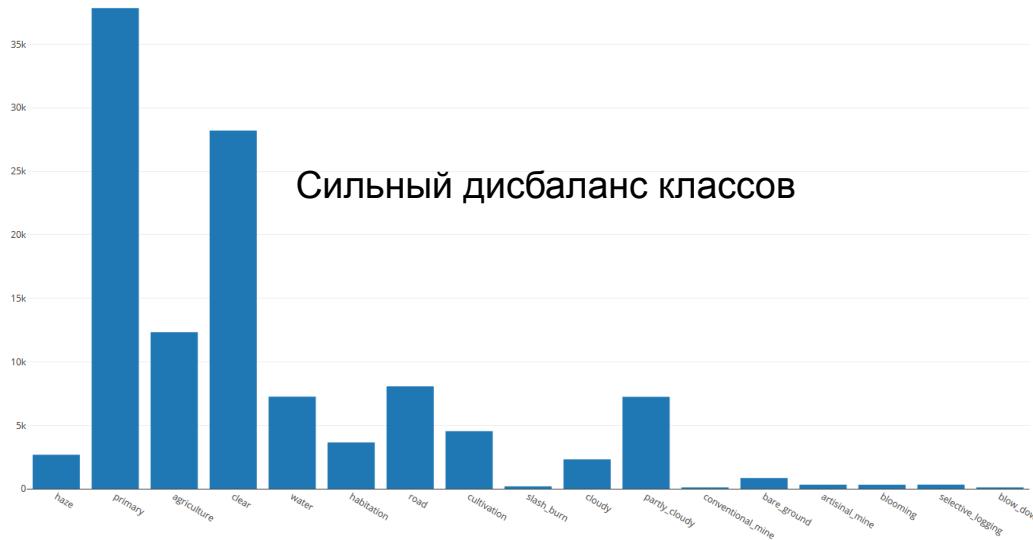
$$(1 + \beta^2) \frac{pr}{\beta^2 p + r} \quad p = \frac{tp}{tp + fp}, \quad r = \frac{tp}{tp + fn}, \quad \beta = 2$$

- Предсказываем не вероятности, а наличие (или отсутствие) соответствующего класса

Примеры изображений для классов

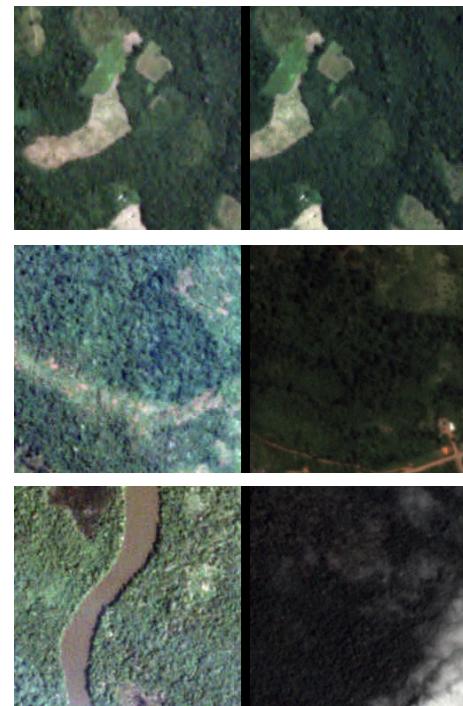
Clear (Ясно)	Partly cloudy (Частично облачно)	Haze (Дымка)	Cloudy (Облачно)	Agriculture (Агрокультура)	Artisinal Mine (Кустарная шахта)
					
Bare Ground (Голая земля)	Blooming (Цветение)	Blow Down (Ветровал)	Conventional Mine (Обычная шахта)	Cultivation (с/х культуры)	Habitation (Поселение)
					
Primary (Типовой лес)	Road (Дороги)	Selective Logging (Выборочная вырубка)	Water (Вода)	Slash Burn (Недавний пожар)	
					

Предварительный анализ и проблемы датасета



- В процессе обнаружился leak - не потерты метаданные. Добавили новых файлов в test, по которым считался Private LB. TIFF для них полностью не соответствовал JPG, позже это поправили.

Несоответствие JPG и TIFF



Обзор решения

Software

Windows 10, Python 3.4 + 3.5

Keras 1.2.1 - основной фреймворк для тренировки нейронных сетей

Theano 0.9.2 - основной бекенд

Tensorflow - доп. бекенд (ResNet152 и Xception)

XGboost, LightGBM, Keras - основные классификаторы

OpenCV - быстрая обработка изображений

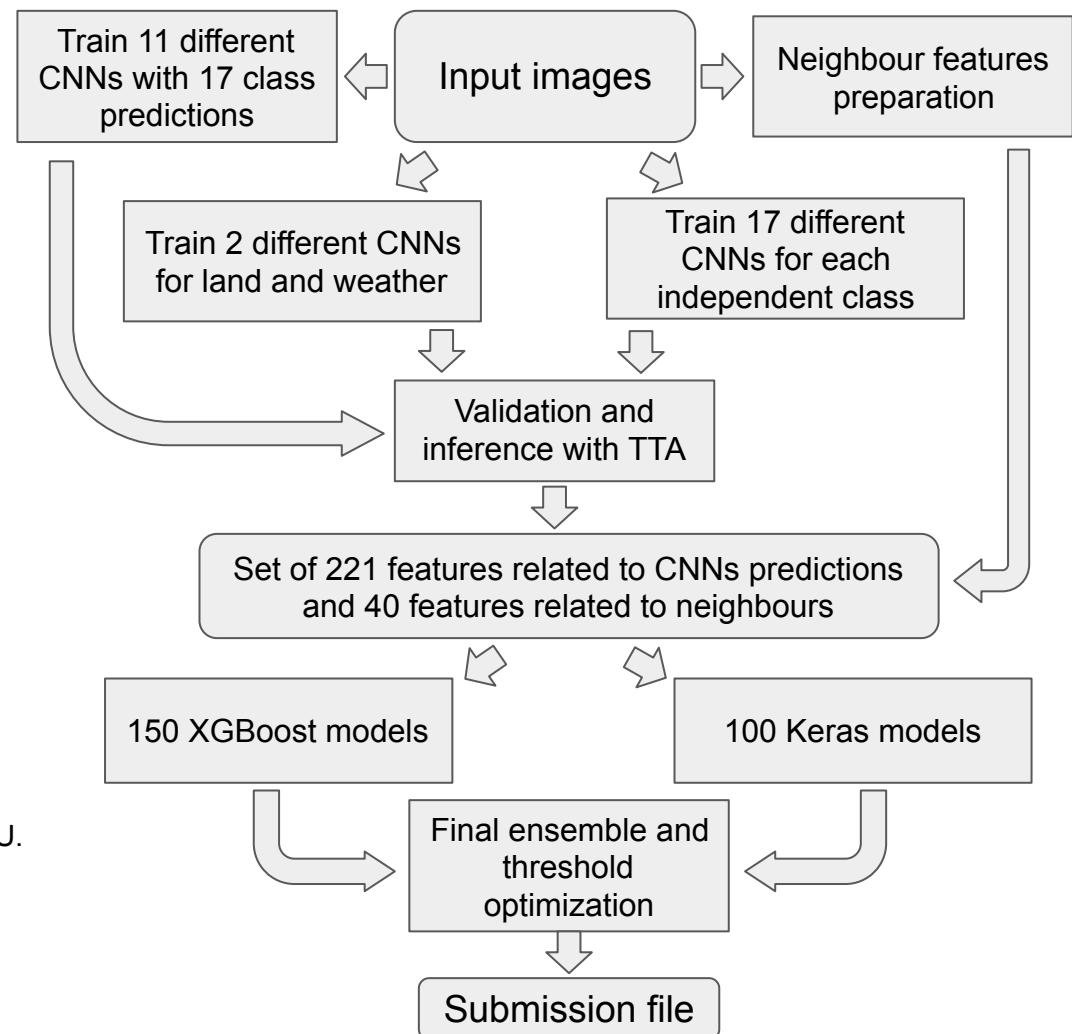
Hardware

6 GPU: 2*1080Ti + 4*1080.

Параллелизация на уровне фолдов или отдельных нейронных сетей.

Время на обучение и обработку данных

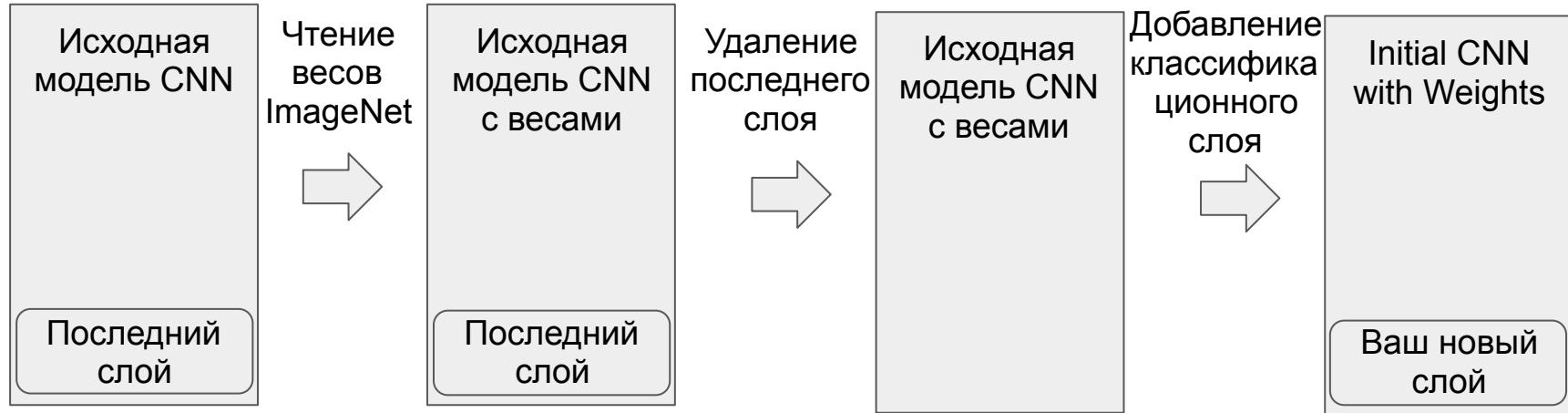
Порядка 1 месяца на одном GPU, около недели на 6 GPU.



Сверточные нейронные сети

CNN Name	Layers	Parameters	Input Shape	Top 1 Error on ImageNet (%)	Top 5 Error on ImageNet (%)
VGG-16	16	134M	224x224	-	7.5
VGG-19	19	139M	224x224	-	7.3
Inception-V3	48	21M	299x299	21.2	5.6
Inception-V4	-	41M	299x299	20.0	5.0
Xception	-	19M	299x299	21.0	5.5
DenseNet-121	121	7M	224x224	25.1	7.8
DenseNet-161	161	27M	224x224	22.4	6.2
DenseNet-169	169	13M	224x224	23.9	6.7
ResNet-50	50	24M	224x224	22.9	6.7
ResNet-101	101	42M	224x224	21.8	6.0
ResNet-152	152	84M	224x224	21.4	5.7

Transfer Learning



- 1) Исходный последний слой - 1000 классов. Заменяем на слой с 17 нейронами для классификации 17 различных классов. Активация Sigmoid, так как изображение может иметь несколько классов.
- 2) Главные плюсы: увеличение скорости обучения, более высокая точность предсказаний.

Набор нейронных сетей в решении

Из-за проблем с TIFF для решения использовались только JPG

- 11 моделей с 17 выходными нейронами и sigmoid активацией на базе разных CNN: VGG16, VGG19, ResNet50, ResNet101, ResNet152, **DenseNet121**, DenseNet169, DenseNet161, Inception v3, Inception v4, Xception
- Одна DenseNet121 модель для погоды: `['clear', 'partly_cloudy', 'haze', 'cloudy']`. Активация “Softmax” поскольку классы взаимоисключающие - 4 выходных нейрона.
- Одна DenseNet121 модель для классов поверхности - 13 выходных нейронов.
- 17 моделей DenseNet121 для каждого отдельного класса. Один выходной нейрон.

Использовалась кроссвалидация 5 KFold. Каждая модель имела 5 наборов весов.

- Всего 30 моделей
- 150 файлов с весами

Training and Data Augmentation

- Модуль Keras для Python с бэкэндом Theano или Tensorflow.
- Типовой размер батча ~20 изображений. 16 изображений для больших сетей из-за ограничений GPU по памяти
- Создание батчей на лету без необходимости хранить всё в памяти: [fit_generator](#)
- Оптимайзер: Adam, learning rate ~0.00003
- Loss function: "logloss". Использование напрямую F2-score давало результат хуже.
- Для моделей с одним классом использовались батчи 50/50, где половина изображений батча имели класс в наличии.

Data augmentation:

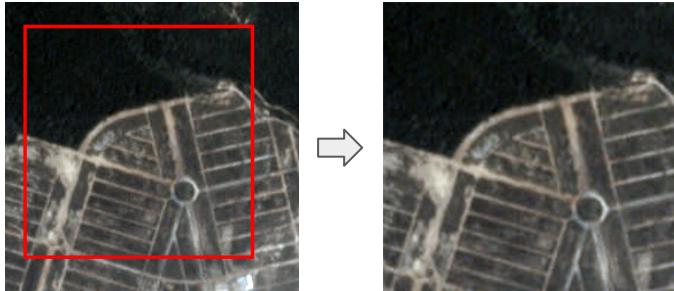
- Random crops: выбрать часть изображения и увеличить до размеров входа CNN (224x224 или 299x299)
- Случайные отражения или вращение на 90 градусов - всего 8 вариантов
- Случайное изменение интенсивности отдельного канала [-10; 10].

Плюсы: значительно увеличивает размер датасета и позволяет использовать test time augmentation (TTA) на этапе обработки тестовых изображений. Фактически в процессе тренировки CNN видит новые изображения в каждом батче.

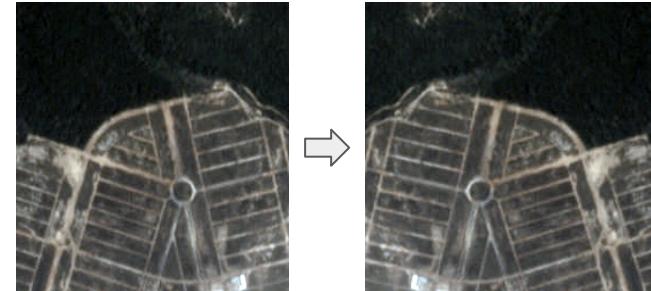
Минусы: При неправильном использовании может сильно замедлить процесс обучения.

Data Augmentation (примеры)

Random crop



Horizontal mirror



90 degree rotation



Color intensity change



Валидация и Test Time Augmentation (TTA)

- 1) Обычная классификация для тренировочных данных: 40К изображений
- 2) Классификация тестовых снимков требует прогон по всем 5 Fold моделям. Таким образом: $60K \times 5 = 300K$ изображений.
- 3) Для увеличения точности классификации мы использовали ТТА. Это означает что каждое изображение с некоторыми изменениями обрабатывается несколько раз. Мы использовали 32 различных варианта для одного изображения. Все 32 предсказания усредняются для получения финального предсказания.
Таким образом одна модель требовала:
 - классификация **1.2M** изображений для тренировочных данных
 - классификация **9.6M** изображений для тестовых данных

По результатам работы получаем два файла, один для тестовых данных, второй для тренировочных с вероятностями по каждому классу. По сути являются числовыми Out of Fold (OOF) фичами для классификаторов второго уровня.

Test Time Augmentation (пример)



Соседние изображения (поиск)

- 1) Очевидная идея что соседние изображения имеют корреляцию между классами
- 2) Мы проверили и выяснили что изображения из тренировочных и обоих тестовых наборов часто являются соседями.
- 3) Поскольку позиции изображений неизвестны, мы попробовали восстановить их алгоритмически

- **Поиск верхнего соседа:**

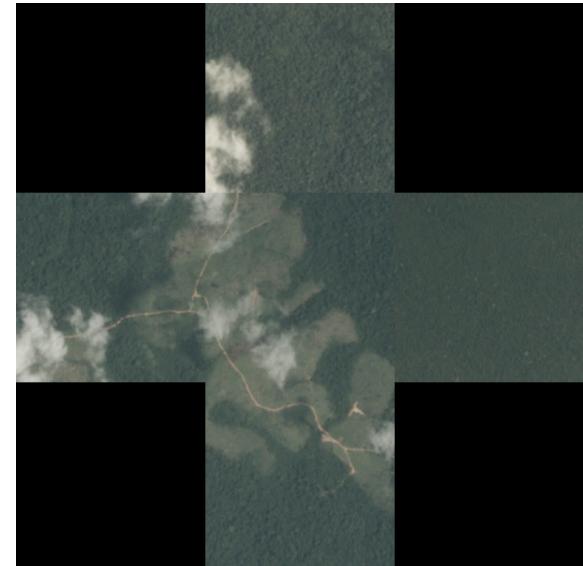
Минимум евклидовой
нормы от разности
векторов верхней
границы изображения и
нижней границы всех
остальных изображений.

- Аналогично для левого,
правого и нижнего
соседей
- Если сосед разный для
TIFF и JPEG, то сосед
найден неверно и мы
можем его удалить.

JPEG соседи для “file_4”

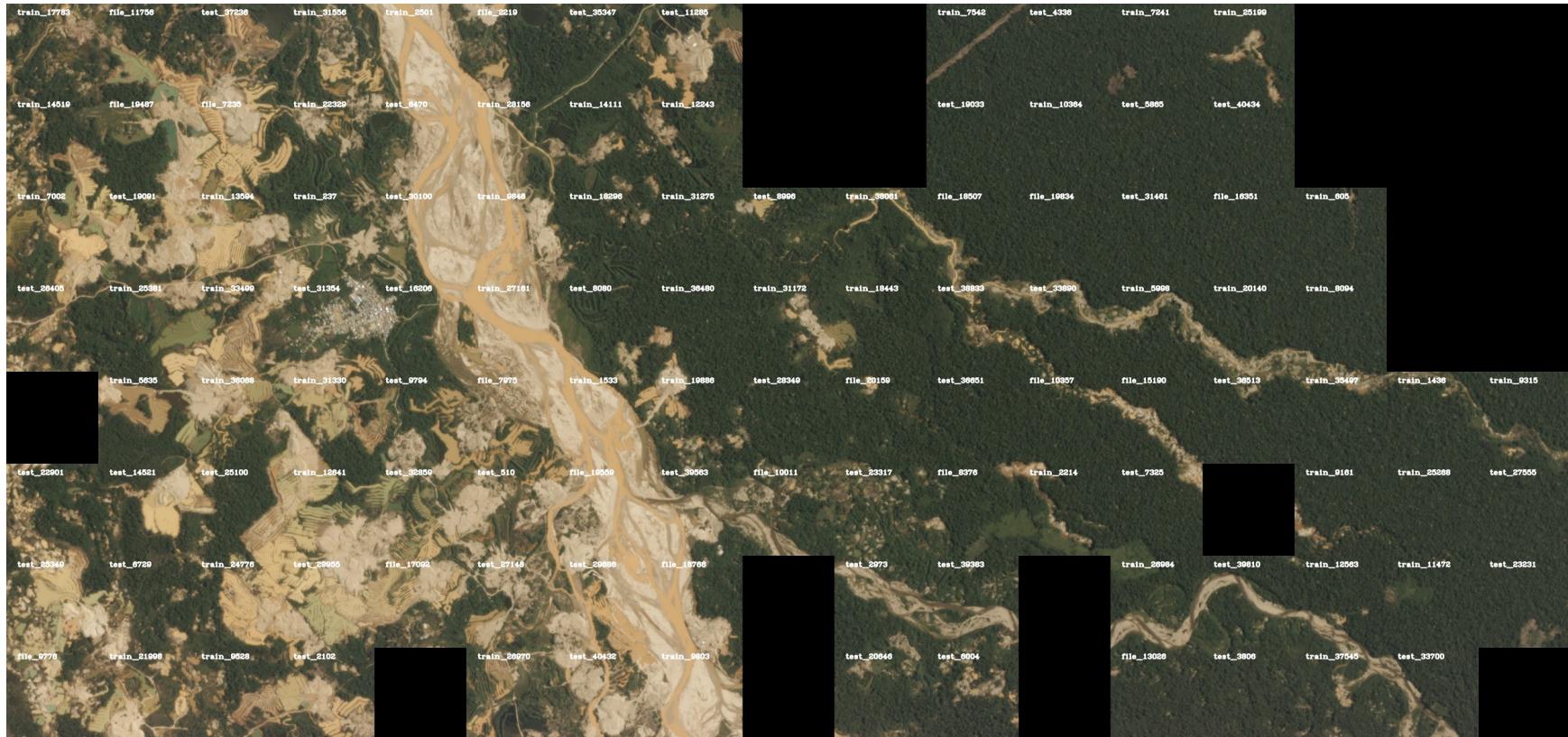


TIFF соседи для “file_4”



Соседние изображения (панорама)

Нам удалось восстановить довольно большие панорамы с помощью нашего метода и присвоить им ID



Соседние изображения (подготовка признаков)

- 1) Усредненные предсказания всех CNN для 4 соседей
- 2) Усредненные предсказания всех CNN для 8 соседей (включая диагональные элементы)
- 3) Усредненные предсказания всех CNN для всей панорамы, которая включает изображение
- 4) Дополнительные фичи для XGBoost: “ID панорамы” и “размер панорамы” основываясь на следующих правилах:
 - панорама содержит более одного элемента
 - панорама содержит хотя бы один элемент из train и хотя бы один из test

Модели второго уровня: XGBoost blender

У нас есть большой набор предсказаний от разных моделей. Нужен метод чтобы объединить их и получить более точные предсказания. Метод который неплохо себя показал - большое число запуска GBM классификатора на разных параметрах с последующим усреднением.

XGboost может предсказать только один класс. Таким образом каждый класс отдельная XGBoost модель на каждый из 17 классов. На входе ~300 признаков.

Что меняем:

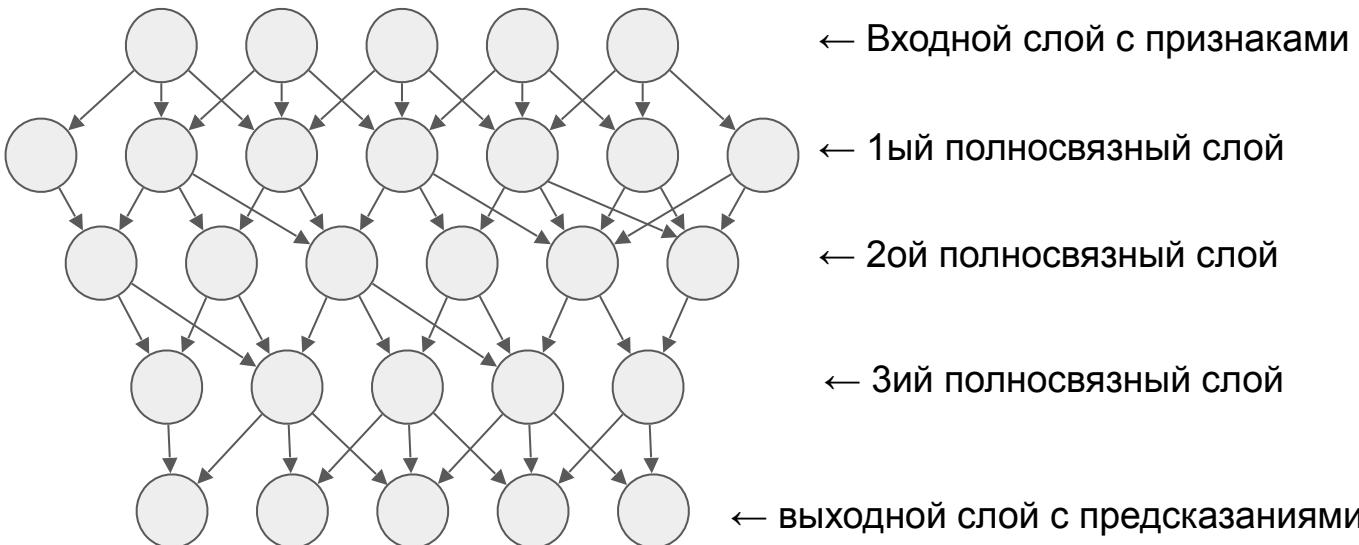
- Folds number [4 - 10]
- Learning rate [0.06 - 0.45]
- Max Depth [2 - 5]
- Subsample [0.6 - 0.99]
- Colsample by tree [0.6 - 0.99]

Модели второго уровня: Keras blender

Keras модели хороши в данном случае:

- 1) Сильно отличаются от моделей XGboost
- 2) Могут предсказывать сразу 17 классов, учитывая взаимодействие между ними

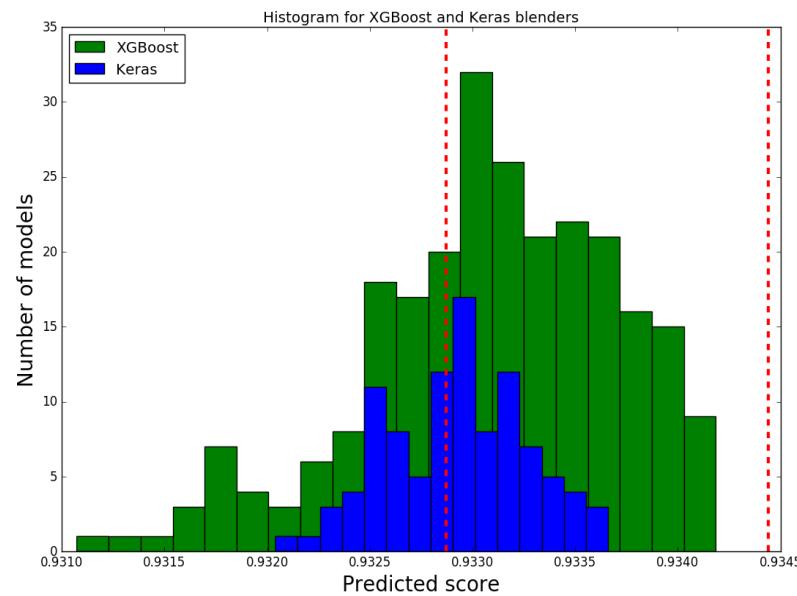
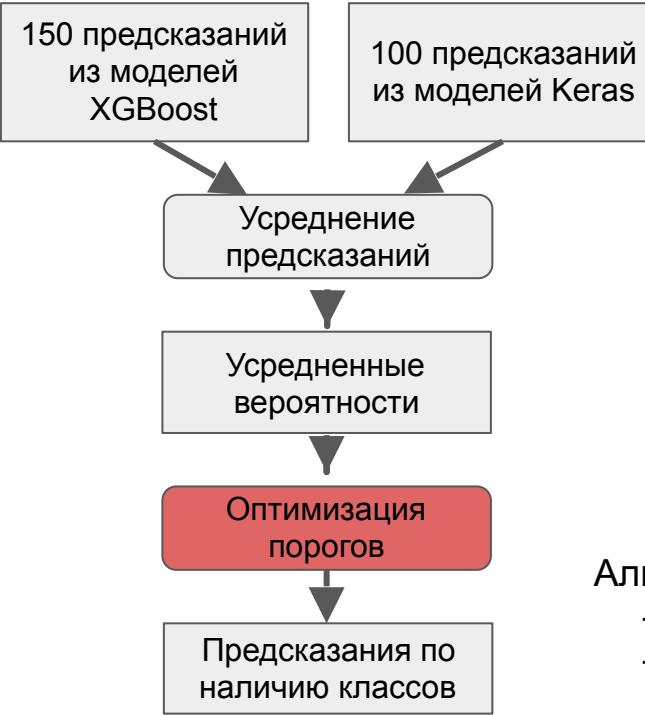
Минус, что точность моделей чуть ниже чем у XGboost или LightGBM.



Параметры для изменения:

- Количество нейронов по уровням
 - 1ый: 400 - 700
 - 2ой: 350 - 600
 - 3ий: 200 - 500
- Значение Dropout
- Случайный выбор активаций: 'RELU', 'ELU', 'PRELU'
- Размер батча: 200-1000
- Learning rate: 1e-5 - 1e-3
- Early stopping: 50 - 150
- Number of folds: 4 - 10

Финальный ансамбль



Алгоритм оптимизации порогов:

- Найти единый порог для всех классов
- Найти оптимальный порог для каждого класса отдельно, зафиксировав остальные. Повторить несколько раз

Пороги для одного из наших лучших сабмитов:

Класс	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Порог	0.224	0.194	0.182	0.261	0.145	0.186	0.097	0.271	0.215	0.235	0.198	0.161	0.183	0.228	0.175	0.164	0.207

Другие эксперименты

- 1) Использование TIFF файлов для обучения Pretrained CNN (7-ми канальный вход). Модификация первого слоя с сохранением весов.
- 2) Обучение с нуля
- 3) Использование F2-score метрики во время обучения
- 4) Предикт сразу нескольких классов в XGBoost блендере
- 5) Ансамбль на базе механизмов голосования множества моделей вместо усреднения

Простая модель для продакшена

Лучшая 5 KFold модель

F2 score: Public: 0.93155 Private: 0.93014

Время обучения: ~12 часов (1 GPU)

Валидация + Классификация 100K изображений: ~12 часов (1 GPU)

Решение 3 место

F2 score: Public: 0.93422 Private: 0.93277

Время обучения: ~15 дней (4-6 GPUs)

Валидация + Классификация 100K изображений: ~15 дней (4-6 GPUs)

Другие упрощения

- 1) Уменьшить число CNN. Оставить только самые точные (или максимально разные)
- 2) Уменьшить число изображений в ТТА с 32 до 8 или даже 2. Ускорение x4 или x16
- 3) Заменить блендеры на оптимизированные одиночные модели

Работа в команде



Google Drive



GitLab

Финальные сабмиты

stas40.csv 2 months ago by Stanislav Semenov Roman (0.5) + Stas (0.5)	0.93302	0.93415	<input type="checkbox"/>
merger_ensemble_by_voting_0.9342354782333031.csv 2 months ago by ZFTurbo Voter ensemble on same seed: Best limit: [68, 72, 34, 46, 51, 51, 47, 28, 41, 36, 73, 65, 41, 48, 60, 20, 42] Best score: 0.9342354782333031	0.93266	0.93448	<input checked="" type="checkbox"/>
merger_0.934460821770783.csv 2 months ago by ZFTurbo CV: 0.93446 - Roman (0.75) + Stas (0.25)	0.93277	0.93421	<input checked="" type="checkbox"/>

Текущий код

Submission and Description	Private Score	Public Score
merger_final_0.9347088497635784.csv a month ago by ZFTurbo XGBoost 150 + Keras 80 Blenders - standard ensemble	0.93302	0.93435

Решение Bestfitting (1 место)

1. Loss-функция: F2 loss в PyTorch
2. Image Haze Removal используя Dark Channel Prior (<http://kaiminghe.com/>)
3. Hard example mining
4. Различные сети с разными размерами входа: 64x64, 224x224, 256x256
5. 17 Ridge regression моделей для каждого класса
6. Ориентировался на CV, а не на LB
7. Не использовал информацию о соседях
8. Лучший сабмит: 9 лучших моделей на CV

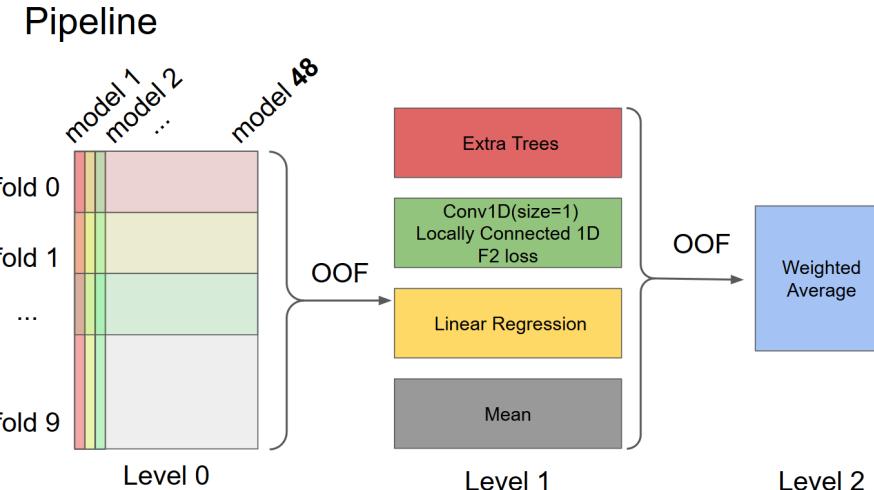


Ссылка: <https://kaggle.com/c/planet-understanding-the-amazon-from-space/discussion/36809>

Решение ODS.ai (7 место)

1. Основной фреймворк: PyTorch
2. 7 участников (> 15 GPU на всех), общие Stratified фолды
3. 48 моделей по 10 фолдов
4. Много разных моделей, включая ResNeXt, DPN и.т.д.
5. Аугментации, TTA, F2 оптимизация по Байесу

	Evgeny Nizhibitsky		Kaggle Master
	Konstantin Lopuhin		Kaggle Master
	Alexander Buslaev		Kaggle Expert
	Vladimir Iglovikov		Kaggle Master
	Alexey Noskov		Kaggle Grandmaster
	n01z3 (you)		Kaggle Master
	Ruslan Baikulov		Kaggle Master



Код решений и другие материалы

Код ZFTurbo: <https://github.com/ZFTurbo/Kaggle-Planet-Understanding-the-Amazon-from-Space>

Код n01z3: https://github.com/N01Z3/kaggle_amazon_from_space

Код kostia: <https://github.com/lopuhin/kaggle-amazon-2017>

Код romul: <https://github.com/lRomul/kaggle-planet-amazon>

Публикации:

1 место: <https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/discussion/36809>

3 место: <https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/discussion/38831>

7 место: <https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/discussion/36955>

Обзор ТОП15:

<https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/discussion/36732>

Выступление от ODS.ai:

<https://youtu.be/8eB1-4ocOKc?t=7645>