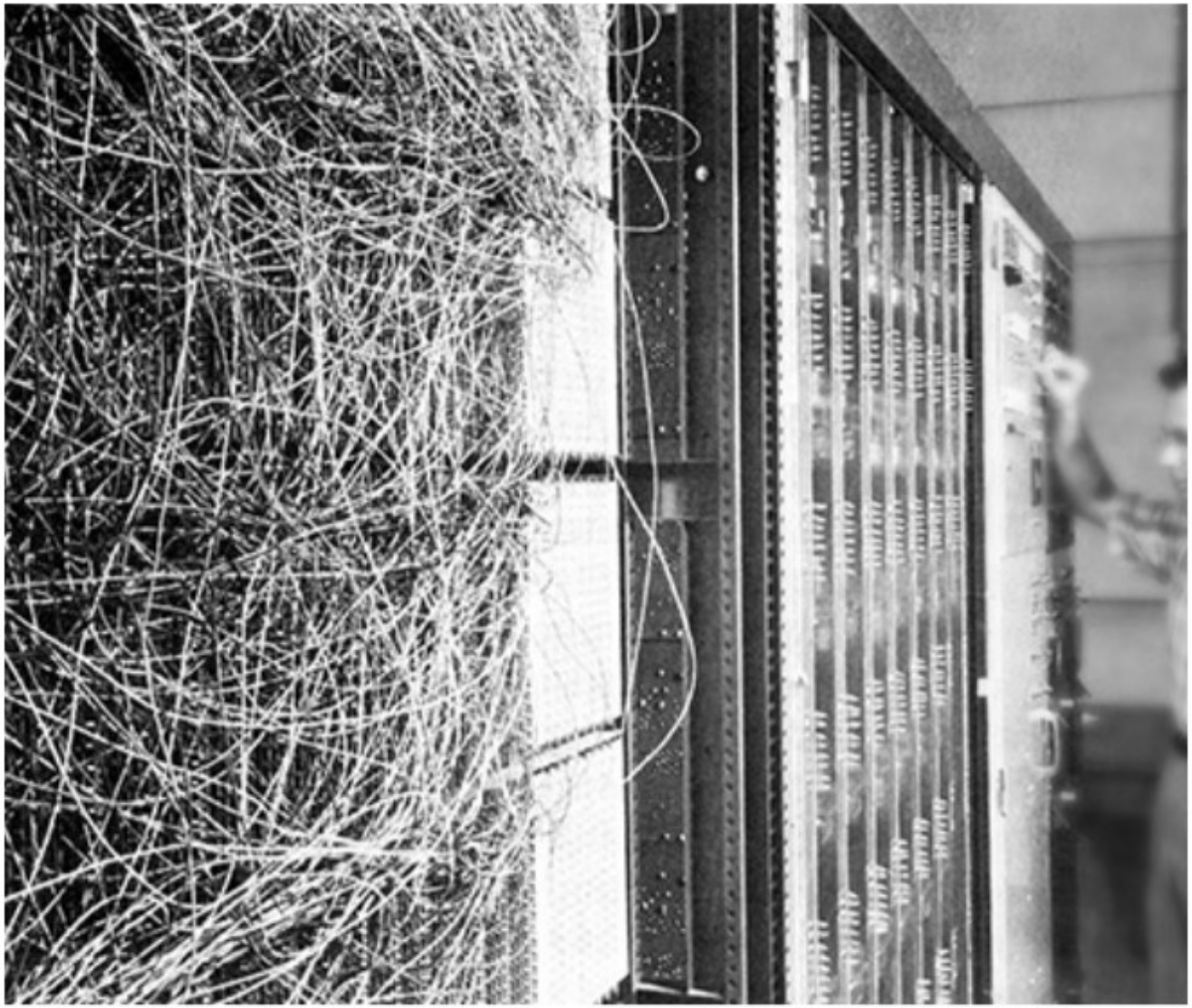
DeepLearning and OpenSource GIS

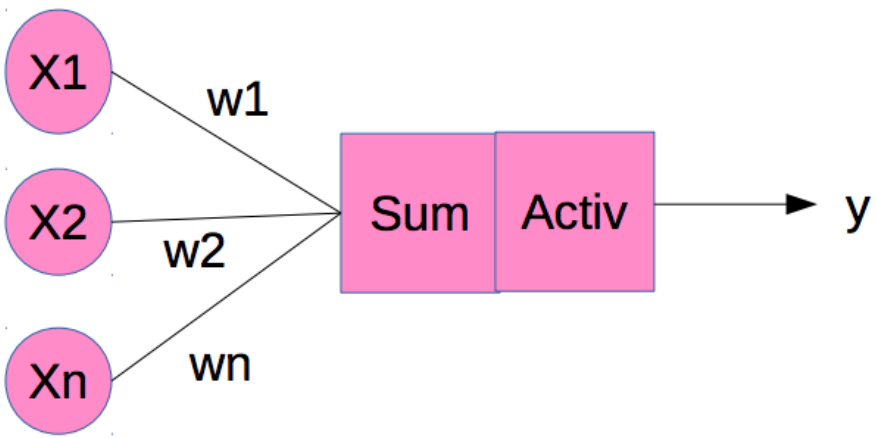
@o_courtin

Forum TIC 2018 - Montpellier

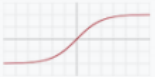
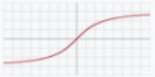
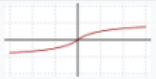


Rosenblatt, The Perceptron (1958)

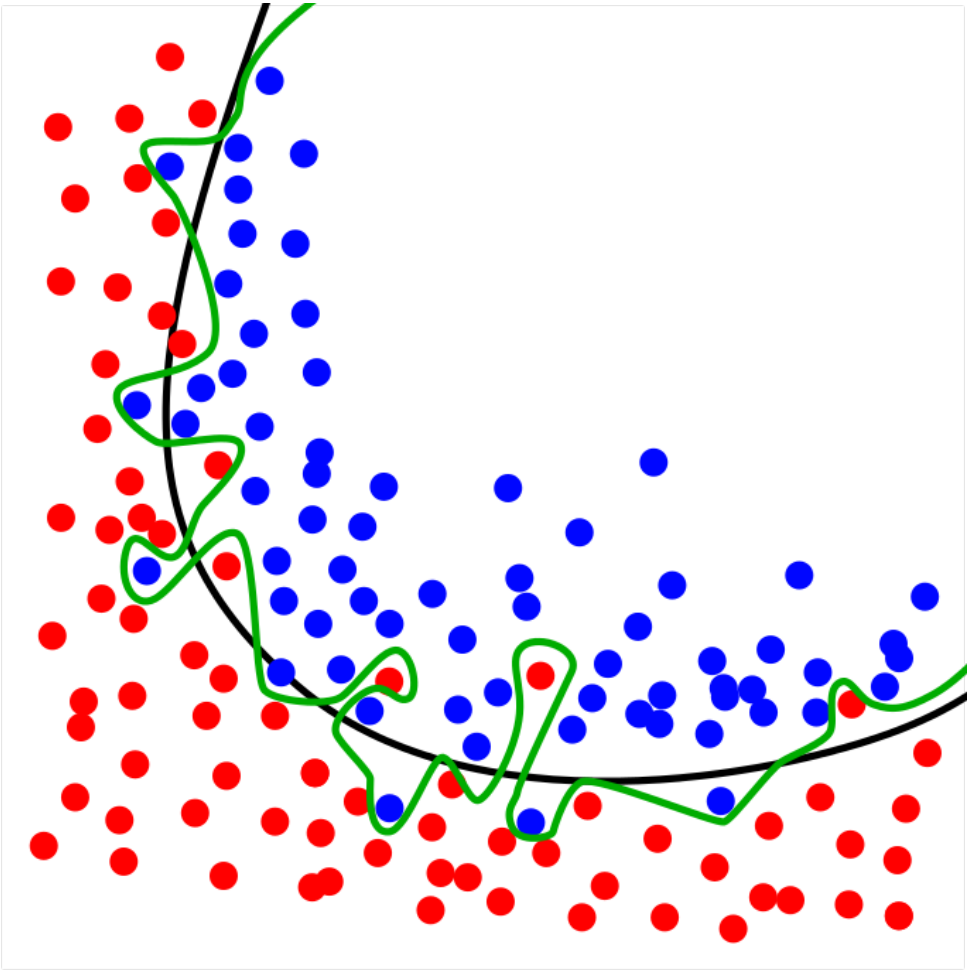
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.335.3398&rep=rep1&type=pdf



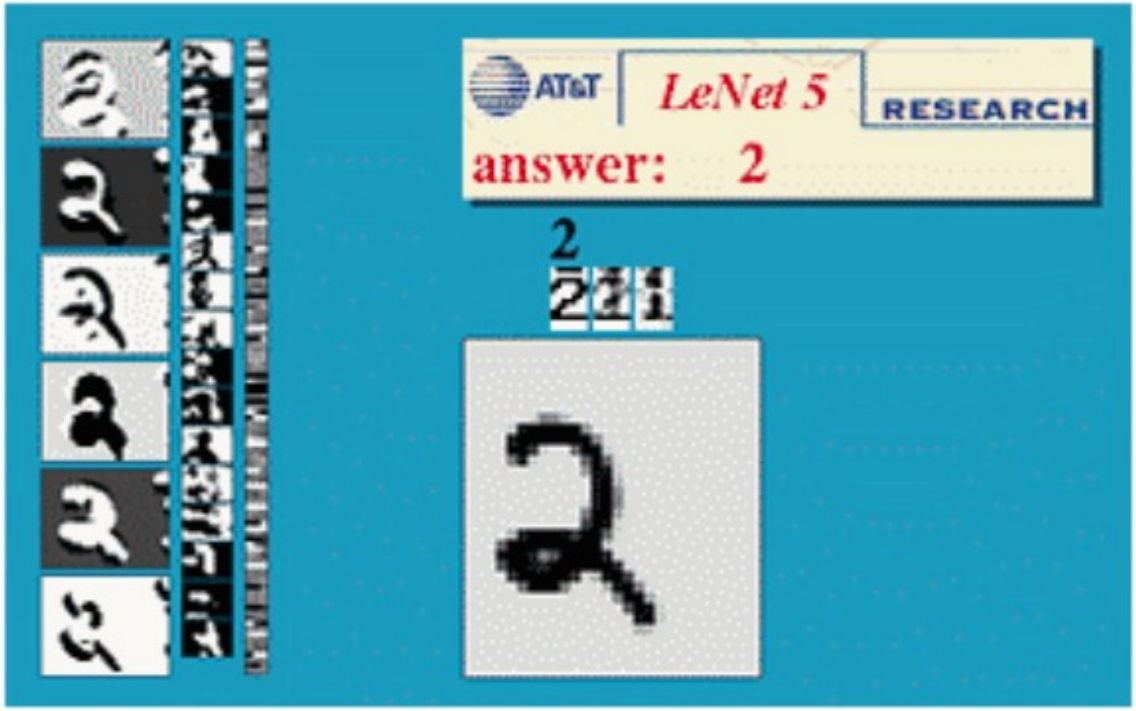
Some activation functions:

Name	Plot	Equation
Identity		$f(x) = x$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Logistic (a.k.a. Soft step)		$f(x) = \frac{1}{1+e^{-x}}$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$
ArcTan		$f(x) = \tan^{-1}(x)$
Softsign [7][8]		$f(x) = \frac{x}{1+\lvert x\rvert}$
Inverse square root unit (ISRU)[9]		$f(x) = \frac{x}{\sqrt{1+\alpha x^2}}$
Rectified linear unit (ReLU)[10]		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$
Leaky rectified linear unit (Leaky ReLU)[11]		$f(x) = \begin{cases} 0.01x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$

Source: Wikipedia

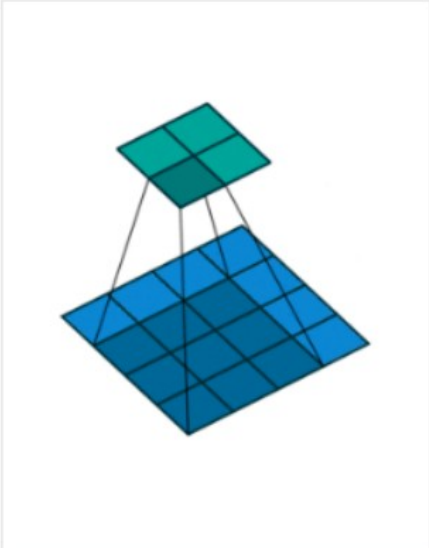


https://en.wikipedia.org/wiki/Overfitting

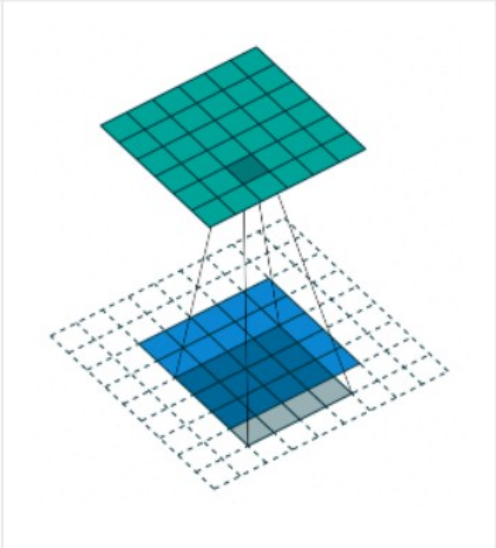


LeCun – LeNet 5 (1998)
http://yann.lecun.com/exdb/lenet/

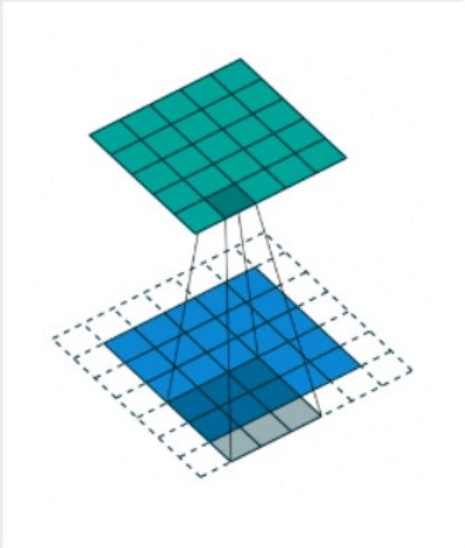
Convolution animations



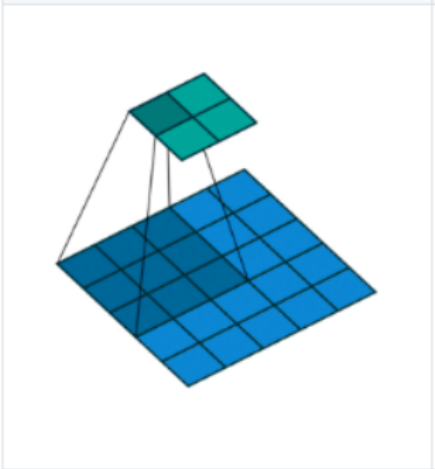
No padding, no strides



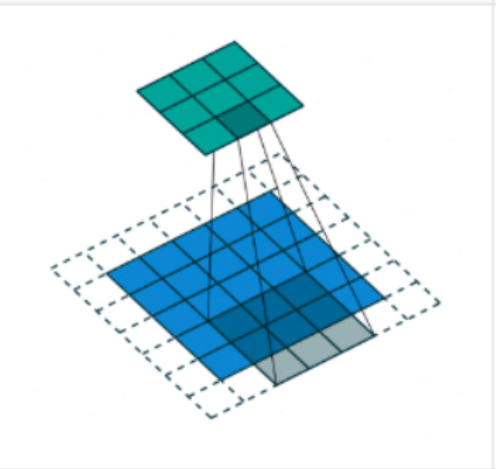
Arbitrary padding, no strides



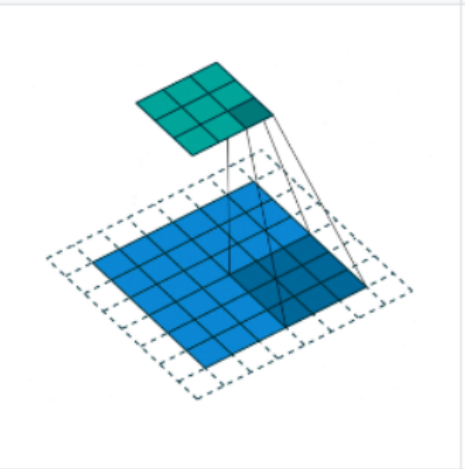
Half padding, no strides



No padding, strides

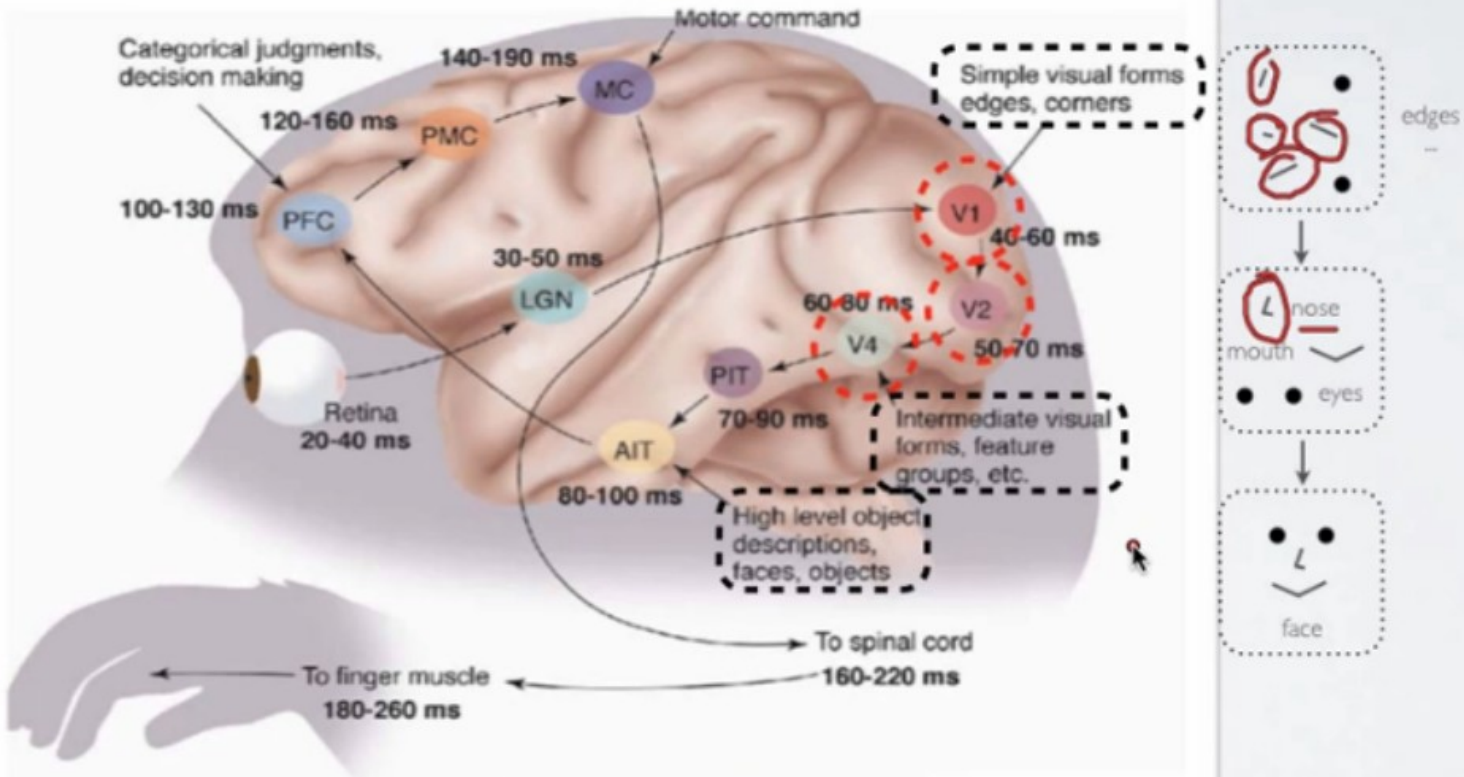


Padding, strides

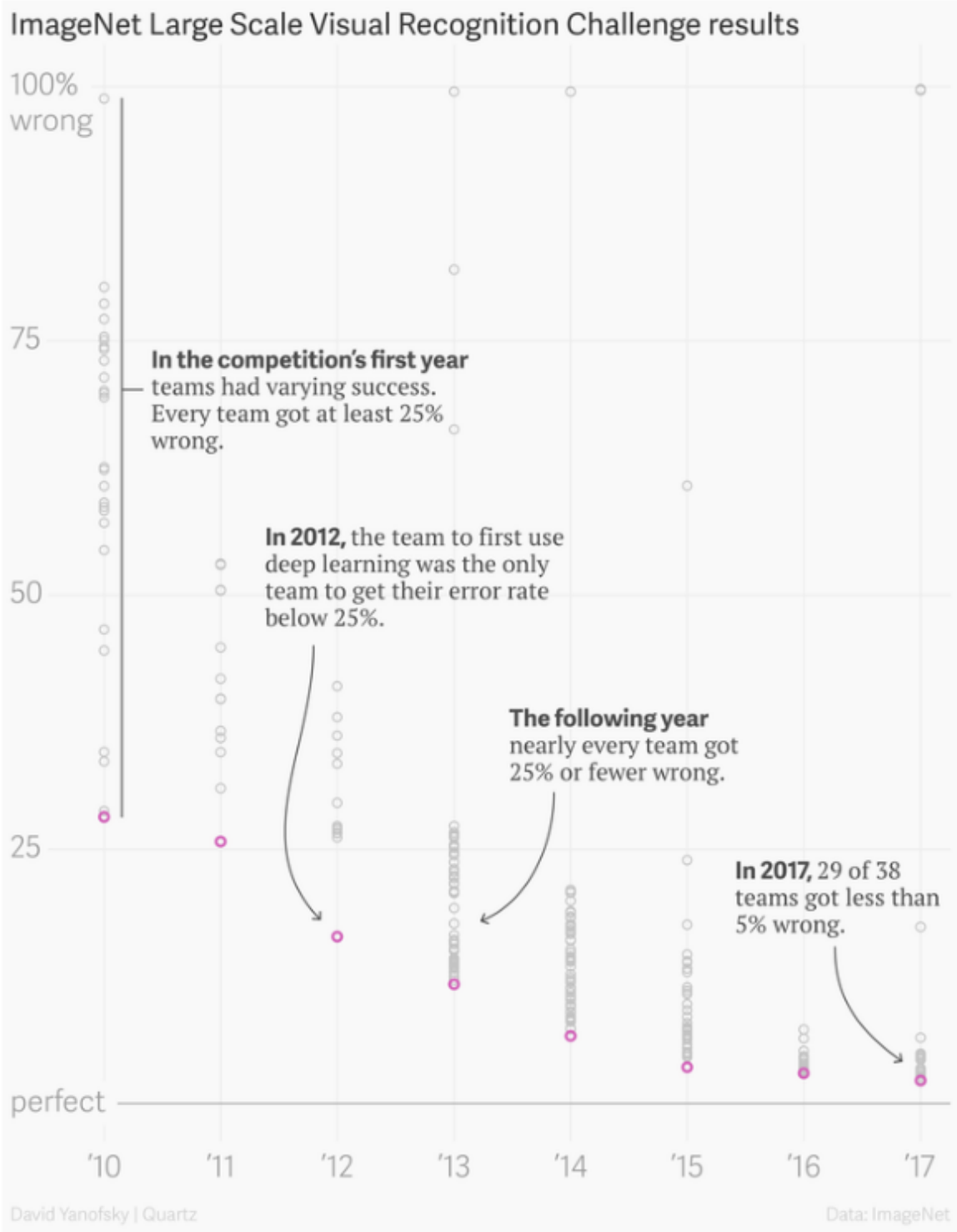


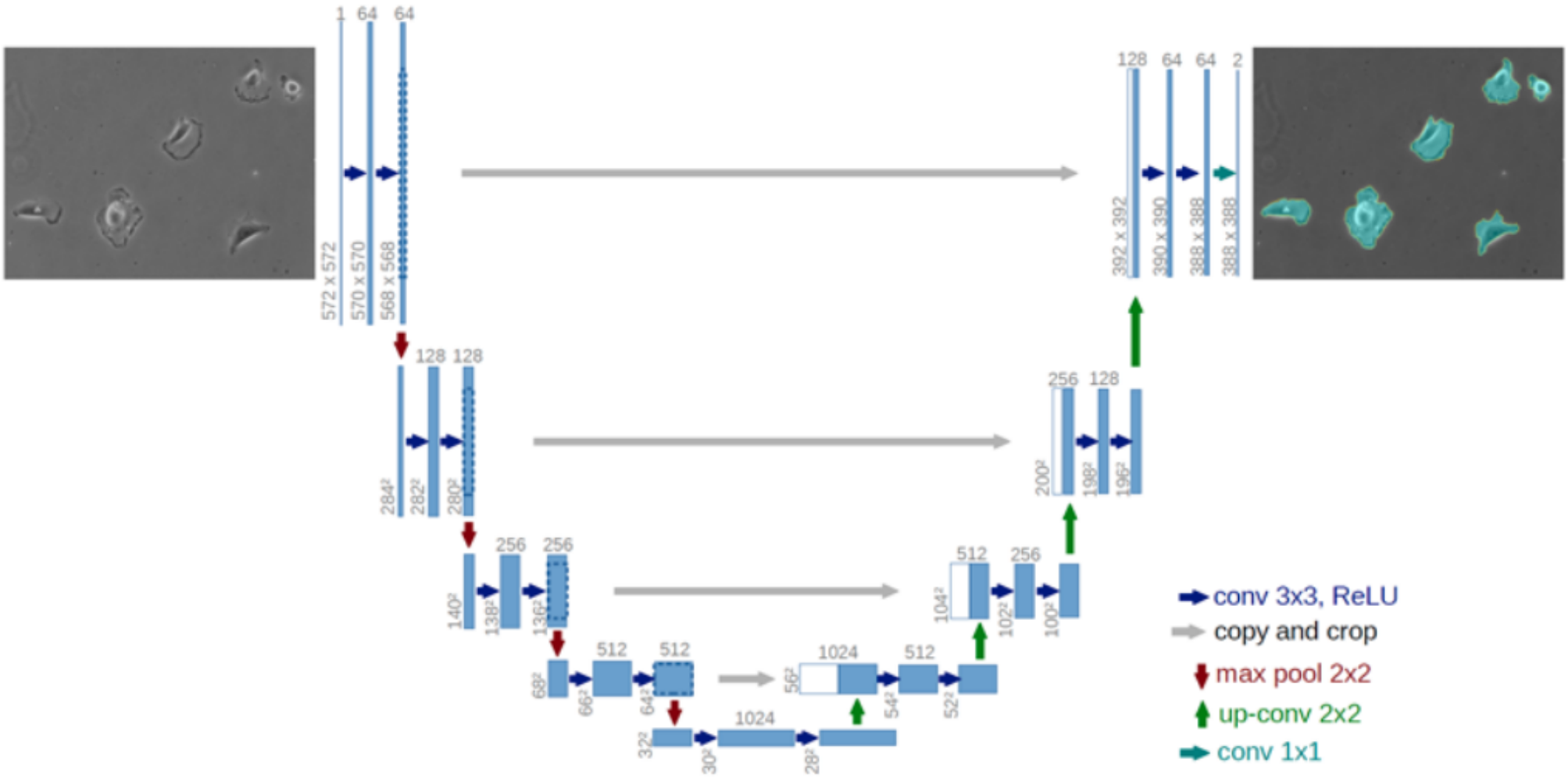
Padding, strides (odd)

https://github.com/vdumoulin/conv_arithmetic



[picture from Simon Thorpe]


ImageNet Large Scale Visual Recognition Challenge results
100% wrong
75
50
25
perfect
'10
'11
'12
'13
'14
'15
'16
'17
In the competition's first year teams had varying success. Every team got at least 25% wrong.
In 2012, the team to first use deep learning was the only team to get their error rate below 25%.
The following year nearly every team got 25% or fewer wrong.
In 2017, 29 of 38 teams got less than 5% wrong.
David Yanofsky | Quartz
Data: ImageNet



U-Net: Convolutional Networks for Biomedical Image Segmentation

https://arxiv.org/abs/1505.04597

IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

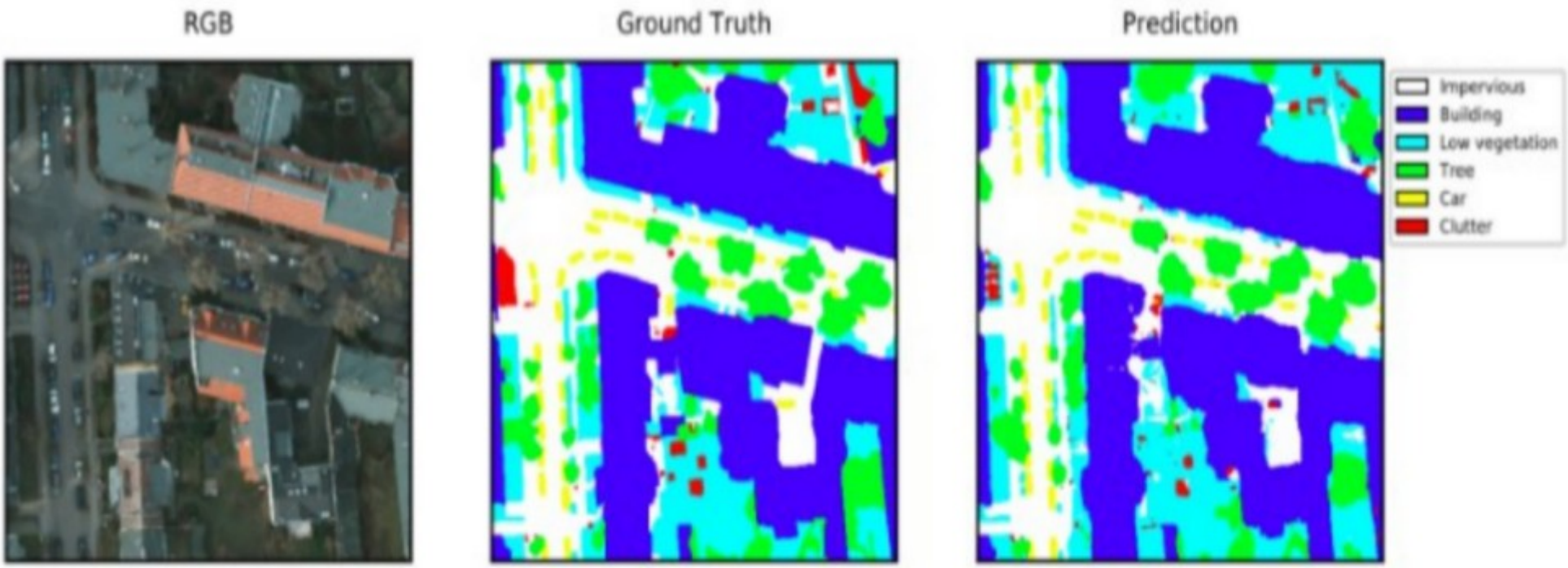
azavea

SOFTWARE DEVELOPMENT

Deep Learning for Semantic Segmentation of Aerial Imagery

By Rob Emanuele on May 30th, 2017

https://www.azavea.com/blog/2017/05/30/deep-learning-on-aerial-imagery/



Pre-trained ResNet50 with ImageNet on IR-R-G

	Overall	Impervious	Building	Low Vegetation	Tree	Car	Clutter
Validation	85.8	89.1	91.8	82.0	83.3	93.7	63.2
Test	89.2	91.4	96.1	86.1	86.6	93.3	46.8

```python
# The number of output labels
nb_labels = 6

# The dimensions of the input images
nb_rows = 256
nb_cols = 256

# A ResNet model with weights from training on ImageNet. This will
# be adapted via graph surgery into an FCN.
base_model = ResNet50(
    include_top=False, weights='imagenet', input_tensor=input_tensor)

# Get final 32x32, 16x16, and 8x8 layers in the original
# ResNet by that layers's name.
x32 = base_model.get_layer('final_32').output
x16 = base_model.get_layer('final_16').output
x8 = base_model.get_layer('final_x8').output

# Compress each skip connection so it has nb_labels channels.
c32 = Convolution2D(nb_labels, (1, 1))(x32)
c16 = Convolution2D(nb_labels, (1, 1))(x16)
c8 = Convolution2D(nb_labels, (1, 1))(x8)

```

```python

# Resize each compressed skip connection using bilinear interpolation.
# This operation isn't built into Keras, so we use a LambdaLayer
# which allows calling a Tensorflow operation.
def resize_bilinear(images):
    return tf.image.resize_bilinear(images, [nb_rows, nb_cols])

r32 = Lambda(resize_bilinear)(c32)
r16 = Lambda(resize_bilinear)(c16)
r8 = Lambda(resize_bilinear)(c8)

# Merge the three layers together using summation.
m = Add()([r32, r16, r8])

# Add softmax layer to get probabilities as output. We need to reshape
# and then un-reshape because Keras expects input to softmax to
# be 2D.
x = Reshape((nb_rows * nb_cols, nb_labels))(m)
x = Activation('softmax')(x)
x = Reshape((nb_rows, nb_cols, nb_labels))(x)

fcn_model = Model(input=input_tensor, output=x)
```

kaggle Search kaggle Competitions Datasets Kernels Discussion Jobs ••• Sign In

[dstl]

Dstl Satellite Imagery Feature Detection

Can you train an eye in the sky?

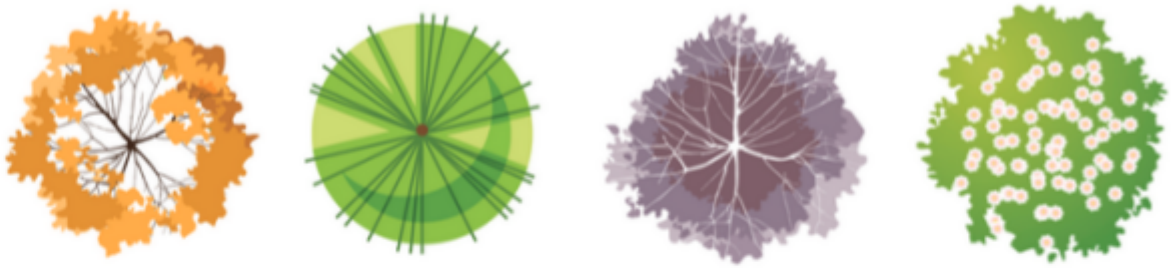
$100,000 · 419 teams · 8 months ago

Overview Data Kernels Discussion Leaderboard Rules

Overview

- Description
- Evaluation
- Prizes
- Data Processing Tutorial
- Timeline

The proliferation of satellite imagery has given us a radically improved understanding of our planet. It has enabled us to better achieve everything from mobilizing resources during disasters to monitoring effects of global warming. What is often taken for granted is that advancements such as these have relied on labeling features of significance like building footprints and roadways fully by hand or through imperfect semi-automated methods.

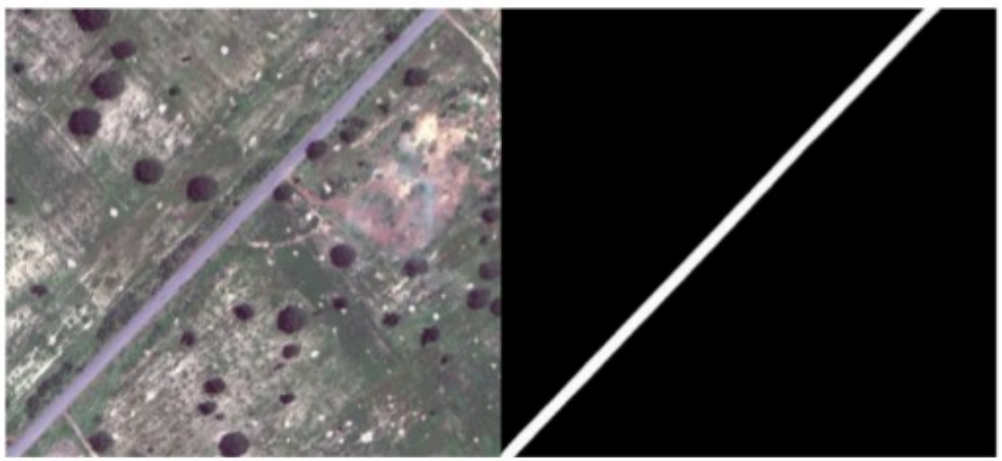


As these large, complex datasets continue to increase exponentially in number, the Defence Science and Technology Laboratory (Dstl) is seeking novel solutions to alleviate the burden on their image analysts. In this competition, Kagglers are challenged to accurately classify features in overhead imagery. Automating feature labeling will not only help Dstl make smart decisions more quickly around the defense and security of the UK, but also bring innovation to computer vision methodologies applied to satellite imagery.

Final results

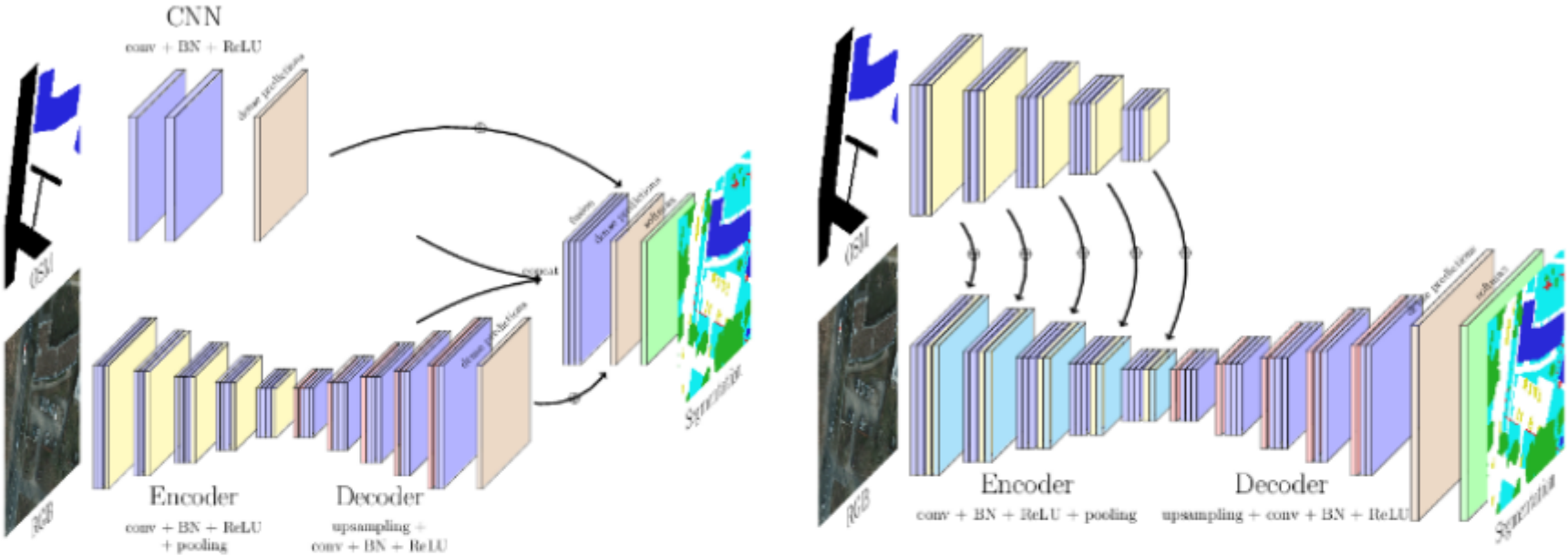
Below we present a small sample of the final results from our models:

Buildings



Type	Wavebands	Pixel resolution	#channels	Size
grayscale	Panchromatic	0.31 m	1	3348 x 3392
3-band	RGB	0.31 m	3	3348 x 3392
16-band	Multispectral	1.24 m	8	837 x 848
	Short-wave infrared	7.5 m	8	134 x 136

https://blog.deepsense.ai/deep-learning-for-satellite-imagery-via-image-segmentation/



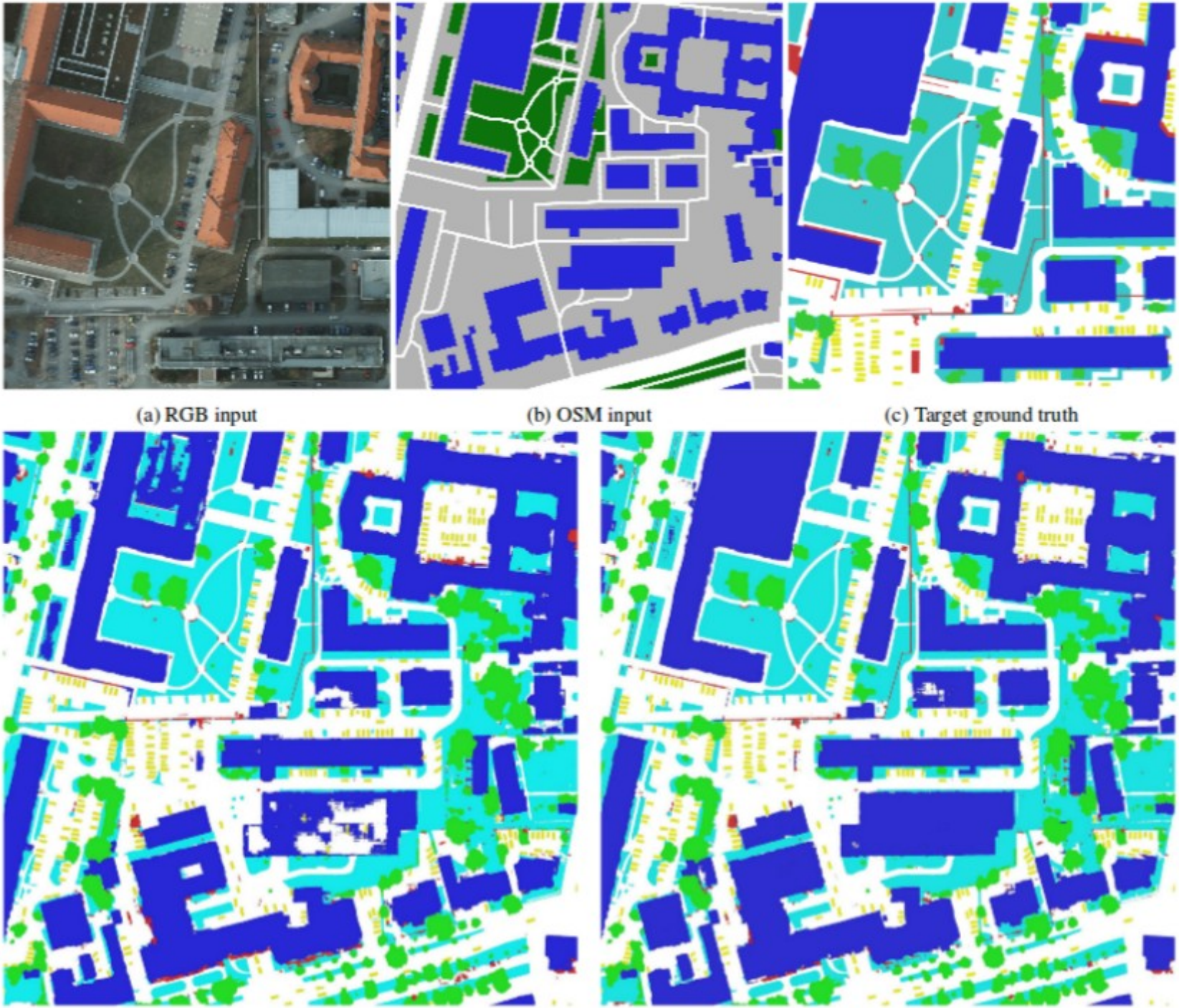
(a) Optical and OSM data fusion using residual correction [1].

(b) FuseNet [13] architecture applied to optical and OSM data.

Figure 1: Deep learning architectures for joint processing of optical and OpenStreetMap data.

Nicolas Audebert, Bertrand Le Saux, Sébastien Lefèvre. Joint Learning from Earth Observation and OpenStreetMap Data to Get Faster Better Semantic Maps. EARTHVISION 2017 IEEE/ISPRS CVPR Workshop. Large Scale Computer Vision for Remote Sensing Imagery, Jul 2017, Honolulu, United States. 2017.

https://hal.archives-ouvertes.fr/hal-01523573



(a) RGB input (b) OSM input (c) Target ground truth

(d) SegNet (RGB) (e) FuseNet (OSM+RGB)

Figure 4: Excerpt from the classification results on Potsdam

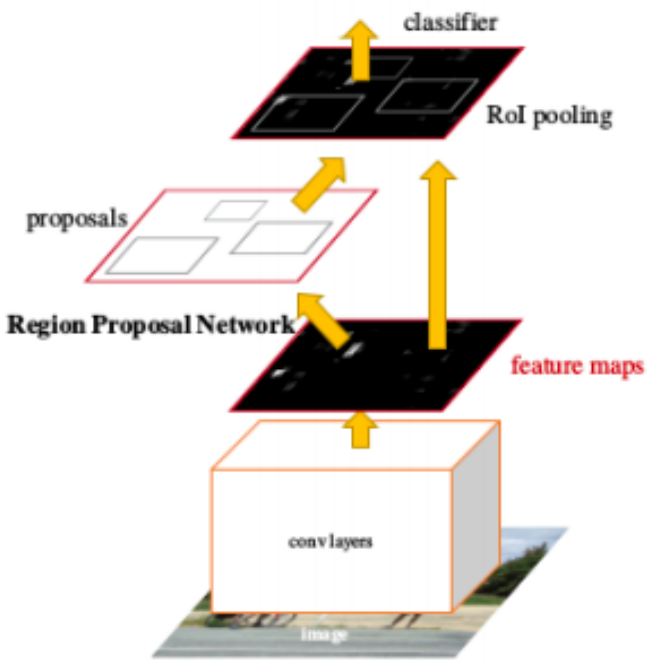


Figure 1. Faster RCNN

https://arxiv.org/pdf/1709.08666.pdf

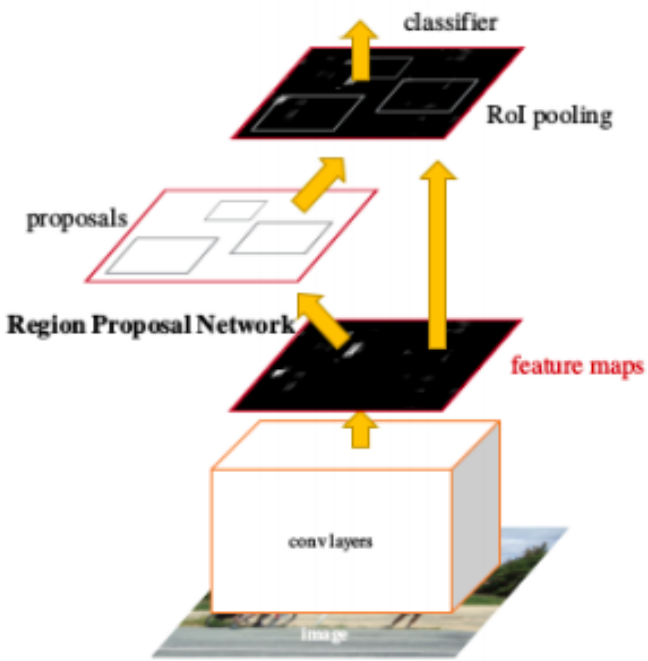
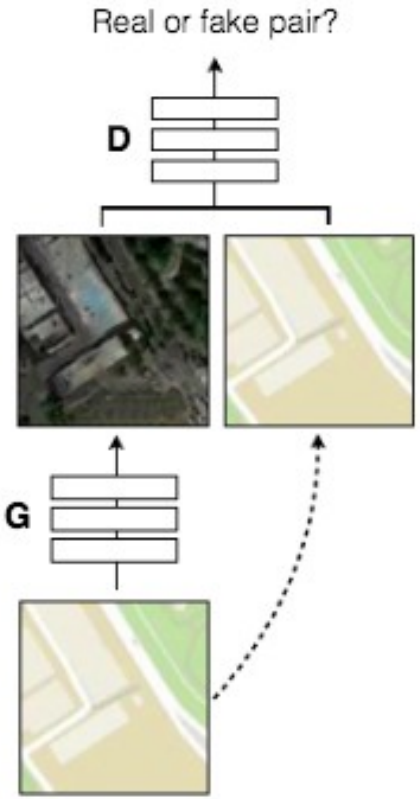


Figure 1. Faster RCNN

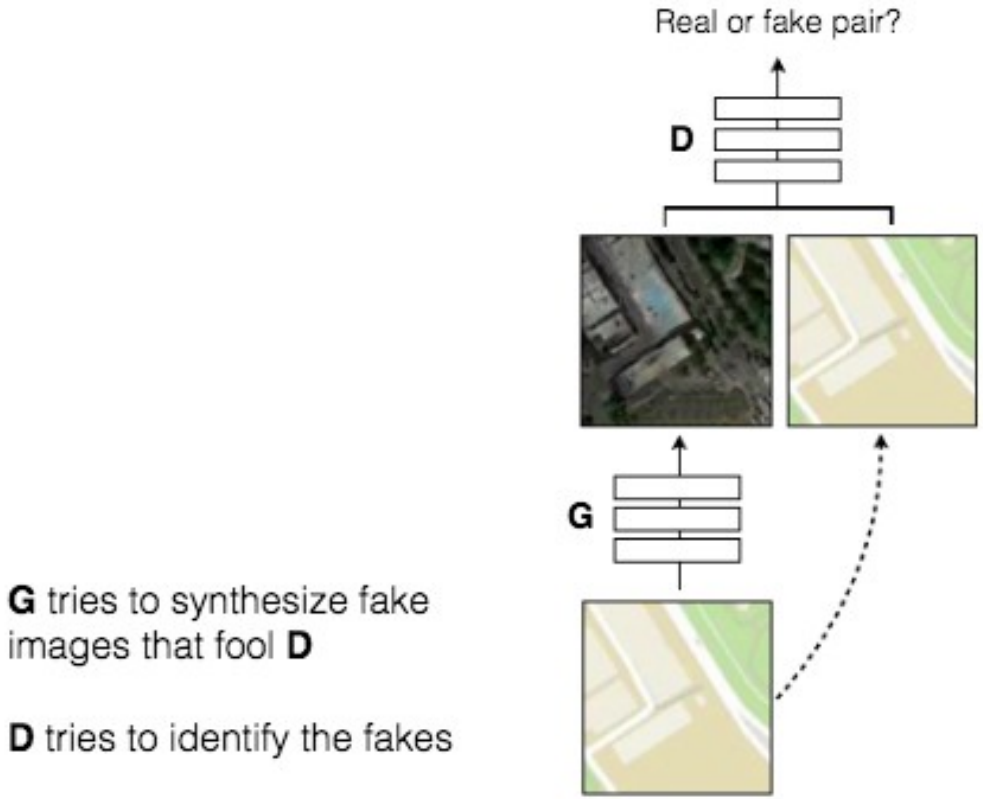
https://arxiv.org/pdf/1709.08666.pdf



G tries to synthesize fake images that fool **D**

D tries to identify the fakes

https://www.researchgate.net/publication/310610633_Image-to-Image_Translation_with_Conditional_Adversarial_Networks



https://www.researchgate.net/publication/310610633_Image-to-Image_Translation_with_Conditional_Adversarial_Networks

Labelled Datasets

Volodymyr PhD: https://www.cs.toronto.edu/%7Evmnih/data/

SpaceNet: https://aws.amazon.com/public-datasets/spacenet/

ISPRS: http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-vaihingen.html
http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-potsdam.html

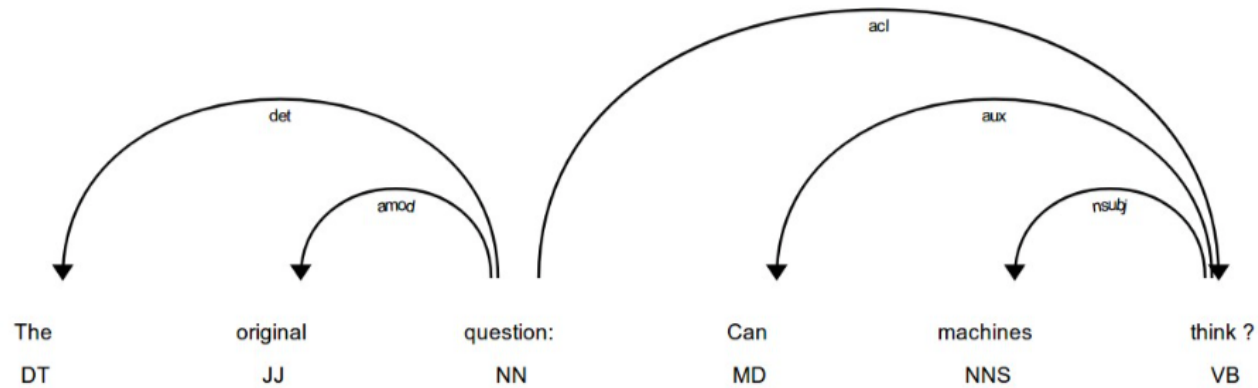
EuroSAT: https://arxiv.org/pdf/1709.00029.pdf

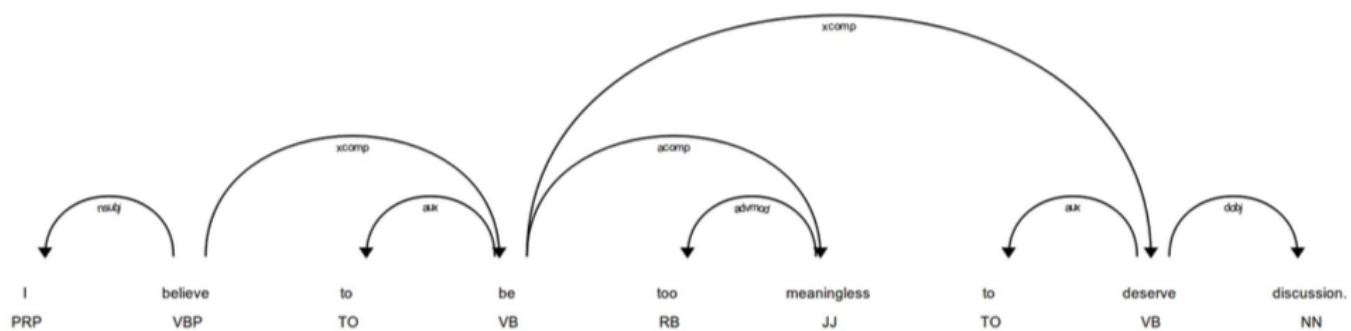
DeepSAT: http://csc.lsu.edu/%7Esaikat/deepsat/

NLP

```python
import spacy
from spacy import displacy

nlp = spacy.load('en')
doc = nlp("""
             The original question: Can Machine think ?
             I believe to be too meaningless to deserve discussion
           """)
displacy.serve(doc, style='dep')
```

```python
import spacy
from spacy import displacy

nlp = spacy.load('en')
doc = nlp("""
            The original question: Can Machine think ?
            I believe to be too meaningless to deserve discussion
          """)
displacy.serve(doc, style='dep')
```





["The Port of Paulsboro is located on the Delaware River and Mantua Creek
in and around Paulsboro, in Gloucester County, New Jersey, US, approximat
ely 78 miles (126 km) from the Atlantic Ocean. Traditionally one of the n
ation's busiest for marine transfer operations, notably for crude oil and
petroleum products, such as jet fuel and asphalt, it is a port of entry w
ith several facilities within a foreign trade zone.\nA part of the port i
s being redeveloped as an adaptable deep water omniport able to handle a
variety of bulk and break bulk cargo, as well as shipping containers. It
is targeted to become a manufacturing/assembly center for wind turbines f
or the development of wind power in New Jersey and other offshore wind po
wer projects along the East Coast of the United States. The Paulsboro Mar
ine Terminal, as it is known, is owned by the South Jersey Port Corporati
on and operated by Holt Logistics. The first ship is expected to arrive a
t the new facility in early 2017 carrying steel for NLMK. The first ship
to call at the port, the Doric Warior, carrying steel for NLMK, arrived M
arch 3, 2017, marking the opening of the new facility."]

https://en.wikipedia.org/wiki/Port_of_Paulsboro

```python
import spacy

nlp = spacy.load('en_core_web_sm')
doc = nlp(text_intro)

for ent in doc.ents:
    print(ent.text, ent.start_char, ent.end_char, ent.label_)
```

```python
import spacy

nlp = spacy.load('en_core_web_sm')
doc = nlp(text_intro)

for ent in doc.ents:
    print(ent.text, ent.start_char, ent.end_char, ent.label_)
```

```
The Port of Paulsboro 2 23 WORK_OF_ART
the Delaware River 38 56 LOC
Mantua Creek 61 73 GPE
Paulsboro 88 97 GPE
Gloucester County 102 119 GPE
New Jersey 121 131 GPE
US 133 135 GPE
approximately 78 miles 137 159 QUANTITY
126 km 161 167 QUANTITY
the Atlantic Ocean 174 192 LOC
New Jersey 695 705 GPE
the East Coast 751 765 LOC
the United States 769 786 GPE
The Paulsboro Marine Terminal 788 817 ORG
the South Jersey Port Corporation 847 880 ORG
Holt Logistics 897 911 ORG
first 917 922 ORDINAL
early 2017 973 983 DATE
NLMK 1003 1007 ORG
first 1013 1018 ORDINAL
the Doric Warior 1045 1061 FAC
NLMK 1082 1086 ORG
March 3, 2017 1096 1109 DATE
```

```python
import spacy

nlp = spacy.load('en_core_web_sm')
doc = nlp(text_intro)

for ent in doc.ents:
    print(ent.text, ent.start_char, ent.end_char, ent.label_)
```

```
The Port of Paulsboro 2 23 WORK_OF_ART
the Delaware River 38 56 LOC
Mantua Creek 61 73 GPE
Paulsboro 88 97 GPE
Gloucester County 102 119 GPE
New Jersey 121 131 GPE
US 133 135 GPE
approximately 78 miles 137 159 QUANTITY
126 km 161 167 QUANTITY
the Atlantic Ocean 174 192 LOC
New Jersey 695 705 GPE
the East Coast 751 765 LOC
the United States 769 786 GPE
The Paulsboro Marine Terminal 788 817 ORG
the South Jersey Port Corporation 847 880 ORG
Holt Logistics 897 911 ORG
first 917 922 ORDINAL
early 2017 973 983 DATE
NLMK 1003 1007 ORG
first 1013 1018 ORDINAL
the Doric Warior 1045 1061 FAC
NLMK 1082 1086 ORG
March 3, 2017 1096 1109 DATE
```

```python
spatial_ent = {}

for ent in doc.ents:
    if (ent.label_ == 'LOC' or ent.label_ == 'GPE'):
        try:
            spatial_ent[ent.text] += 1
        except:
            spatial_ent[ent.text] = 1

print(spatial_ent)
```

```
{'the East Coast': 1, 'the Atlantic Ocean': 1, 'New Jersey': 2, 'the United States': 1, 'the
Delaware River': 1, 'Mantua Creek': 1, 'Paulsboro': 1, 'US': 1, 'Gloucester County': 1}
```

NLP extraction + Geoname Gazeeter + Classification

```python
from mordecai import Geoparser
geo = Geoparser()
```

```python
res = geo.geoparse(text_intro)

import pprint as pp
pp.pprint(res)
```

```python
from mordecai import Geoparser
geo = Geoparser()
```

```python
res = geo.geoparse(text_intro)

import pprint as pp
pp.pprint(res)
```

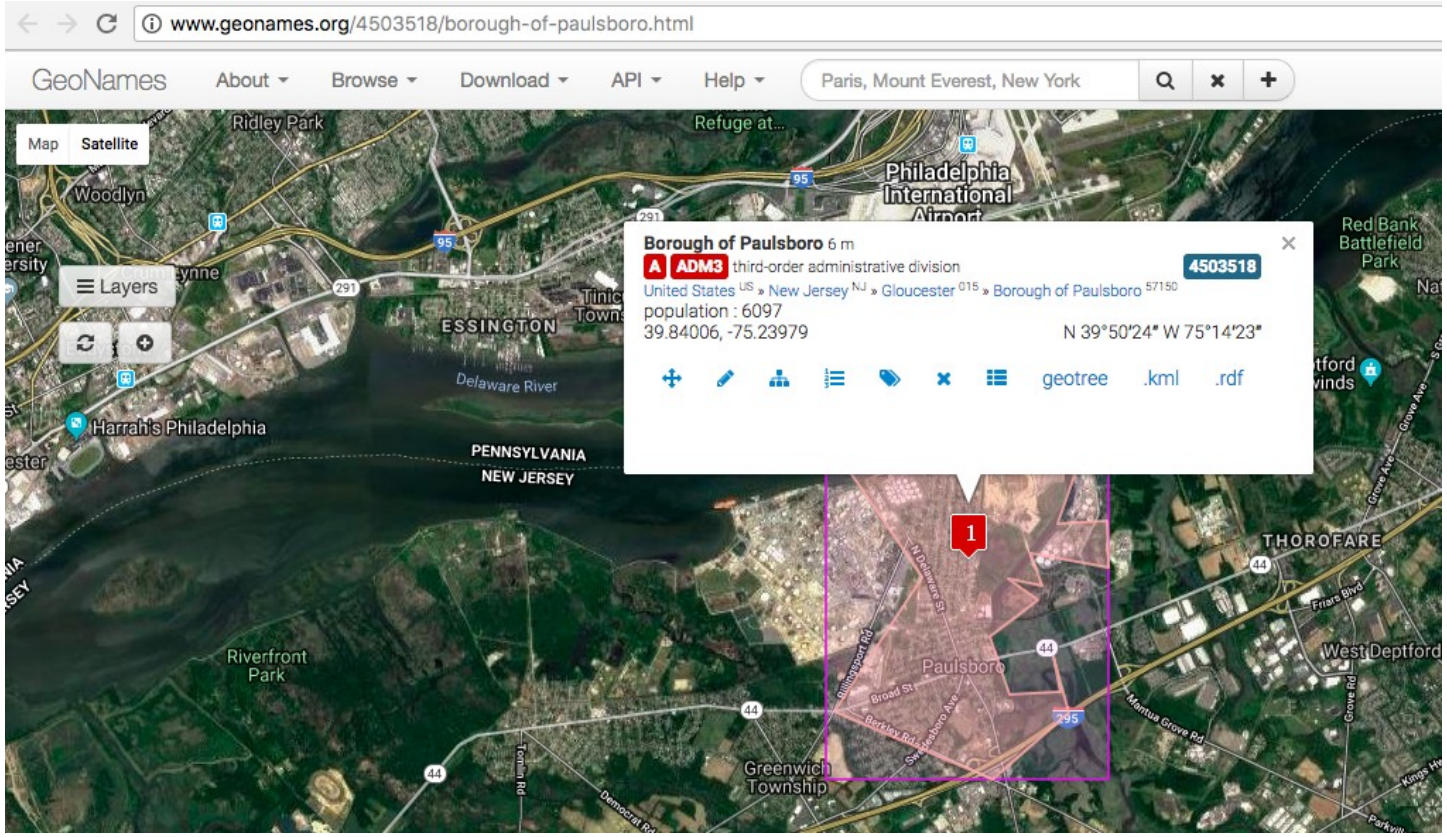
```
[{'country_conf': 0.9991698,
  'country_predicted': 'USA',
  'geo': {'admin1': 'Delaware',
          'country_code3': 'USA',
          'feature_class': 'A',
          'feature_code': 'ADM1',
          'geonameid': '4142224',
          'lat': '39.00039',
          'lon': '-75.49992',
          'place_name': 'Delaware'},
  'spans': [{'end': 8, 'start': 0}],
  'word': 'Delaware'},
 {'country_conf': 0.999931,
  'country_predicted': 'USA',
  'geo': {'admin1': 'New Jersey',
          'country_code3': 'USA',
          'feature_class': 'H',
          'feature_code': 'STM',
          'geonameid': '4502882',
          'lat': '39.85286',
          'lon': '-75.23013',
          'place_name': 'Mantua Creek'},
  'spans': [{'end': 12, 'start': 0}],
  'word': 'Mantua Creek'},
 {'country_conf': 0.9992422,
  'country_predicted': 'USA',
  'geo': {'admin1': 'New Jersey',
          'country_code3': 'USA',
          'feature_class': 'A',
          'feature_code': 'ADM3',
          'geonameid': '4503518',
          'lat': '39.84006',
          'lon': '-75.23979',
          'place_name': 'Borough of Paulsboro'},
  'spans': [{'end': 9, 'start': 0}],
  'word': 'Paulsboro'},
 {'country_conf': 0.9950605,
  'country_predicted': 'USA',
  'geo': {'admin1': 'New Jersey',
          'country_code3': 'USA',
          'feature_class': 'A',
          'feature_code': 'ADM2',
          'geonameid': '4501944',
```

```
for r in res:
    if r['country_conf'] > 0.9:
        print(r['geo']['place_name'],
              "http://www.geonames.org/{}/".format(r['geo']['geonameid']))
```

```
Delaware http://www.geonames.org/4142224/
Mantua Creek http://www.geonames.org/4502882/
Borough of Paulsboro http://www.geonames.org/4503518/
Gloucester County http://www.geonames.org/4501944/
New Jersey http://www.geonames.org/5101760/
Atlantic Ocean Palm Inn http://www.geonames.org/6528913/
New Jersey http://www.geonames.org/5101760/
East Coast Baptist Church http://www.geonames.org/7238976/
United States http://www.geonames.org/6252001/
```

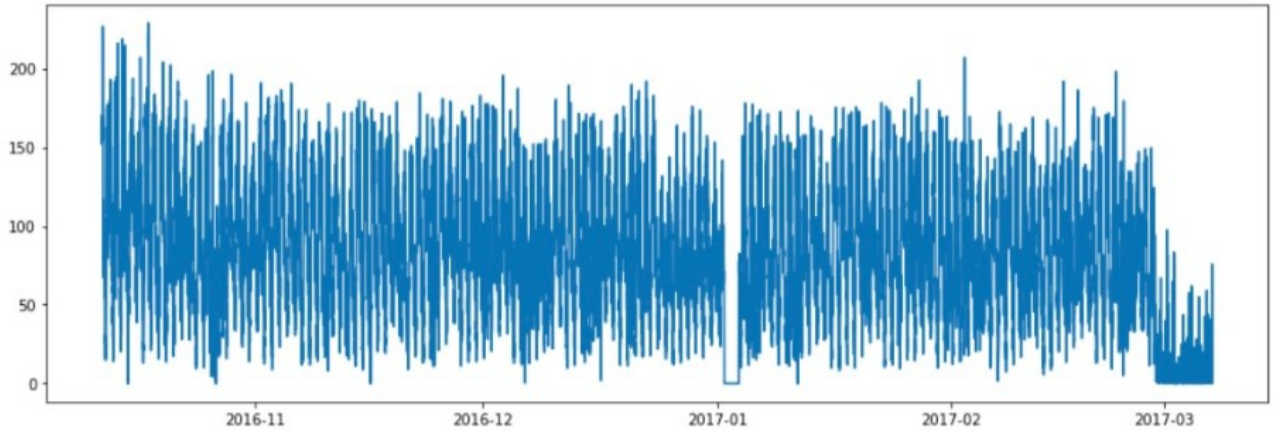
```python
for r in res:
    if r['country_conf'] > 0.9:
        print(r['geo']['place_name'],
              "http://www.geonames.org/{}/".format(r['geo']['geonameid']))
```

```
Delaware http://www.geonames.org/4142224/
Mantua Creek http://www.geonames.org/4502882/
Borough of Paulsboro http://www.geonames.org/4503518/
Gloucester County http://www.geonames.org/4501944/
New Jersey http://www.geonames.org/5101760/
Atlantic Ocean Palm Inn http://www.geonames.org/6528913/
New Jersey http://www.geonames.org/5101760/
East Coast Baptist Church http://www.geonames.org/7238976/
United States http://www.geonames.org/6252001/
```



Time Series


https://opendata.paris.fr/explore/dataset/place-de-la-nation-points-de-mesure-flux/map/?location=18,48.84852,2.39599
PARISDATA
MAIRIE DE PARIS
Les données
L'API
La licence
La démarche
Cartographe
66 enregistrements
Aucun filtre actif.
Filtres
Rechercher...
type
turnstile 50
area 16
classes 2
ped 47
vehicle 17
bike 2
classes 3
bike 21
ped 1
Place de la Nation - Points de mesure flux
Informations
Tableau
Carte
Analyse
Export
API
id
7 118
nom
Cercle interieur - niveau Fabre d'Eglantine
metrics
[{"id": "direction_1", "name": "Sens inverse"}, {"id": "direction_2", "name": "Sens de circulation"}]
classes
all
type
turnstile
classes 2
vehicle
classes 3
bike
Avenue de Taillebourg
Avenue du Trône
Boulevard Diderot
Place de la Nation
Rue de Picpus
Lycée général et technologique Aragó
Nation
RER A
Métro 1
Métro 2
Métro 9
30 m


200
150
100
50
0
2016-11
2016-12
2017-01
2017-02
2017-03

facebook / **prophet**

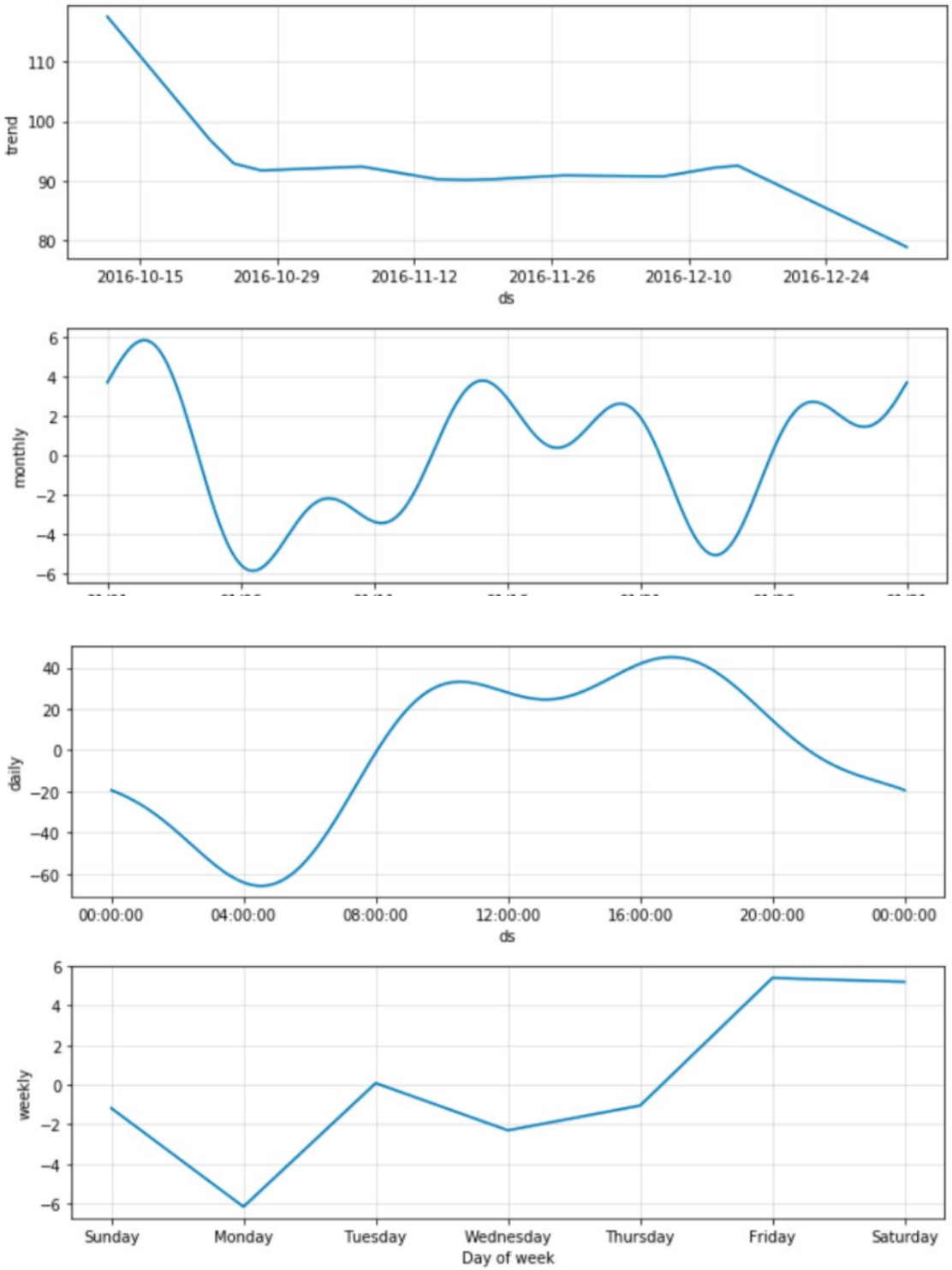
Watch 226 | Star 4,114 | Fork 728

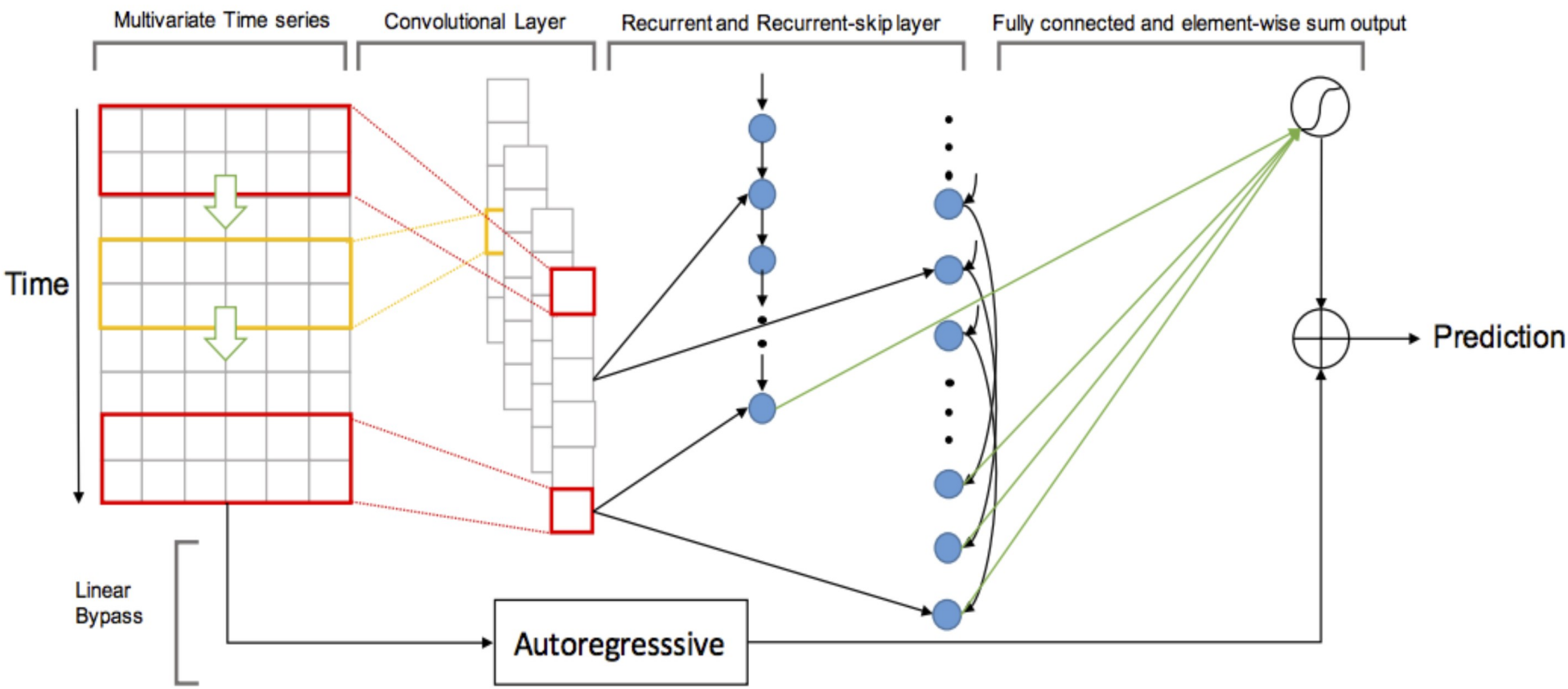
Prophet: Automatic Forecasting Procedure

Prophet is a procedure for forecasting time series data. It is based on an additive model where non-linear trends are fit with yearly and weekly seasonality, plus holidays. It works best with daily periodicity data with at least one year of historical data. Prophet is robust to missing data, shifts in the trend, and large outliers.

Prophet is open source software released by Facebook's Core Data Science team. It is available for download on CRAN and PyPI.

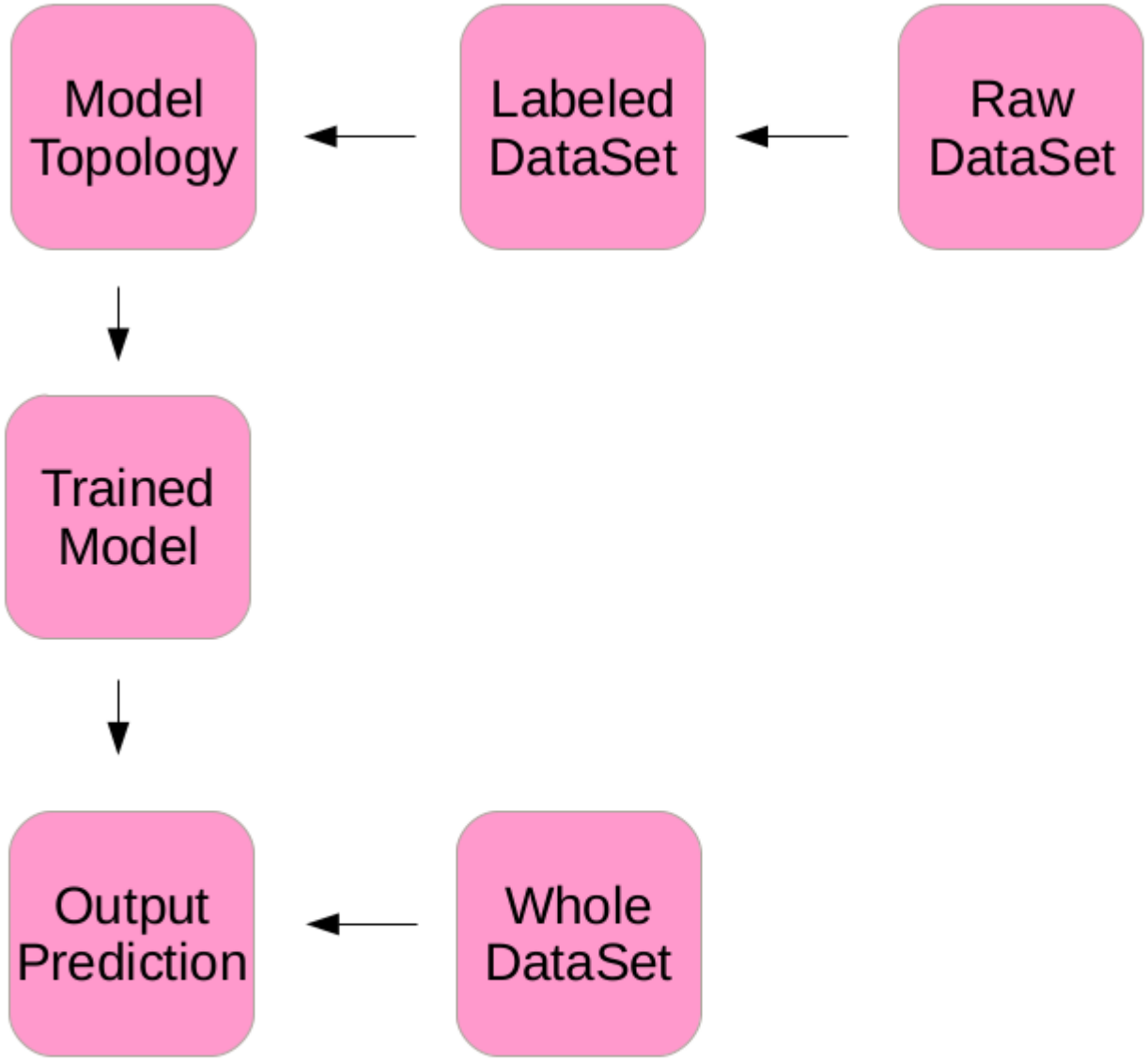
https://github.com/facebook/prophet

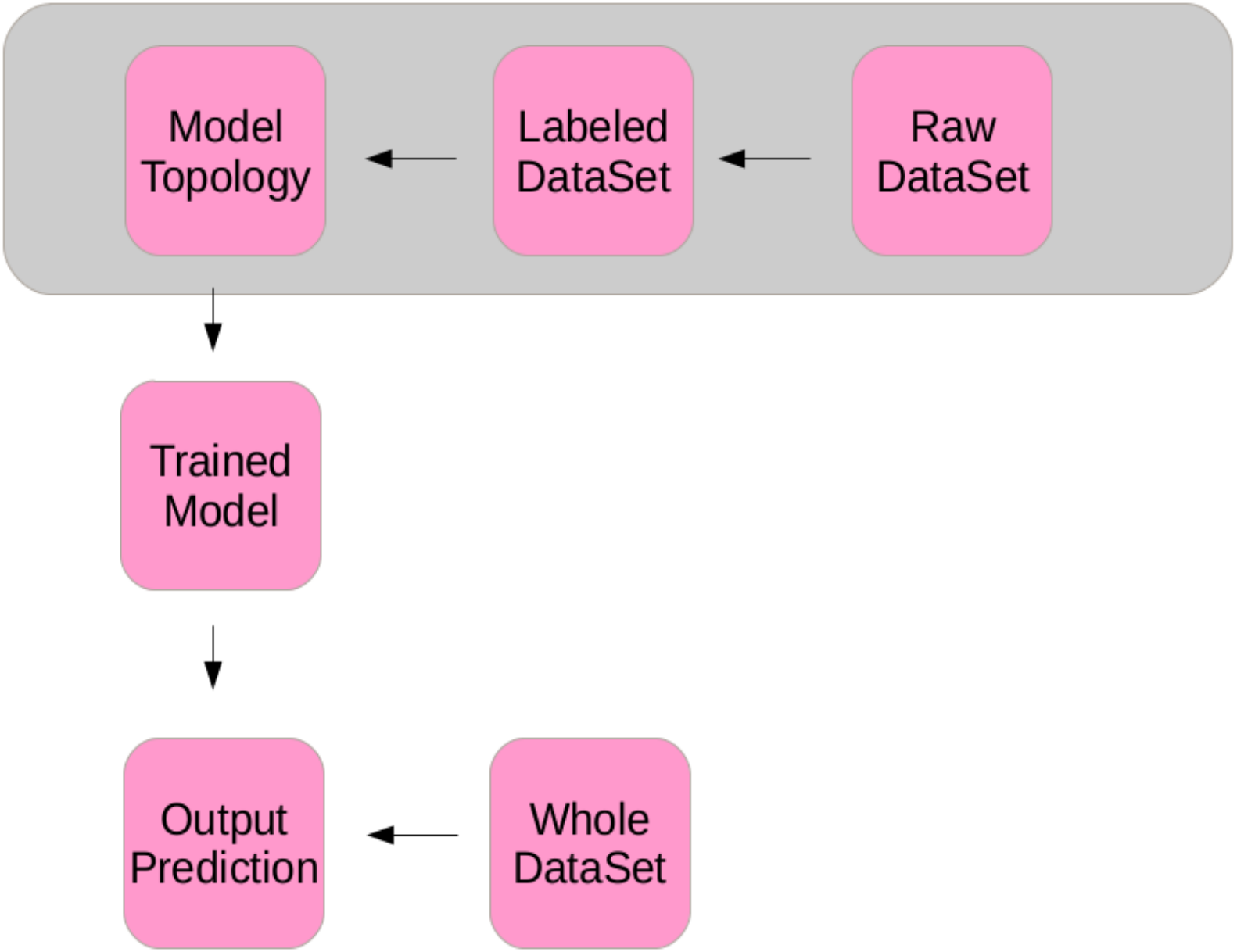

trend
110
100
90
80
2016-10-15
2016-10-29
2016-11-12
2016-11-26
2016-12-10
2016-12-24
ds
monthly
6
4
2
0
−2
−4
−6
daily
40
20
0
−20
−40
−60
00:00:00
04:00:00
08:00:00
12:00:00
16:00:00
20:00:00
00:00:00
ds
weekly
6
4
2
0
−2
−4
−6
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Day of week

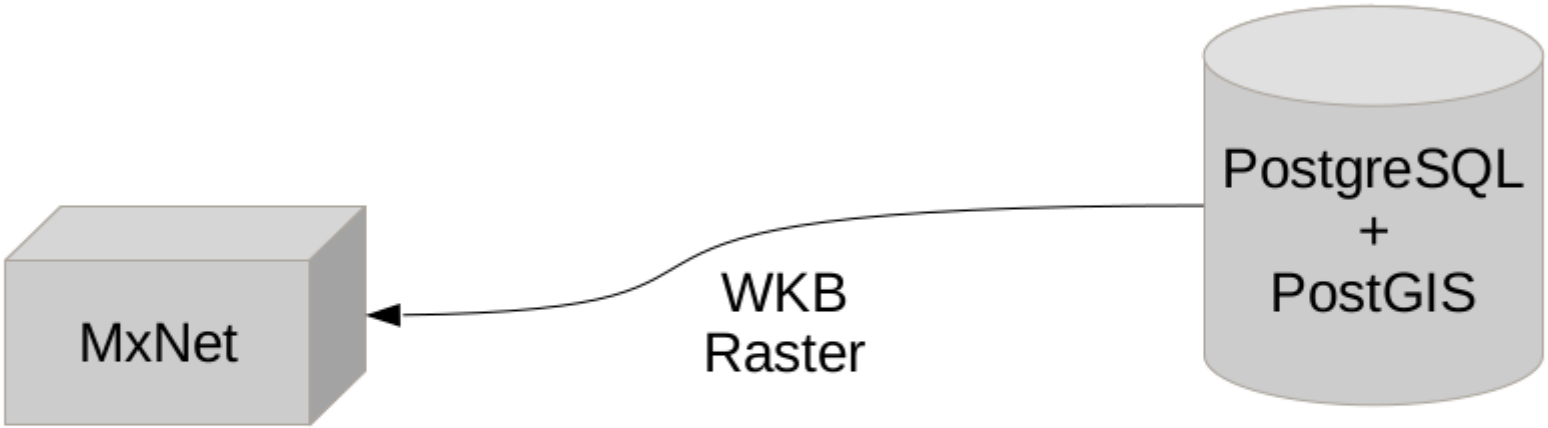


https://arxiv.org/abs/1703.07015


THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS


Model Topology
Labeled DataSet
Raw DataSet
Trained Model
Output Prediction
Whole DataSet


Model Topology
Labeled DataSet
Raw DataSet
Trained Model
Output Prediction
Whole DataSet


PostgreSQL
+
PostGIS
WKB
Raster
MxNet

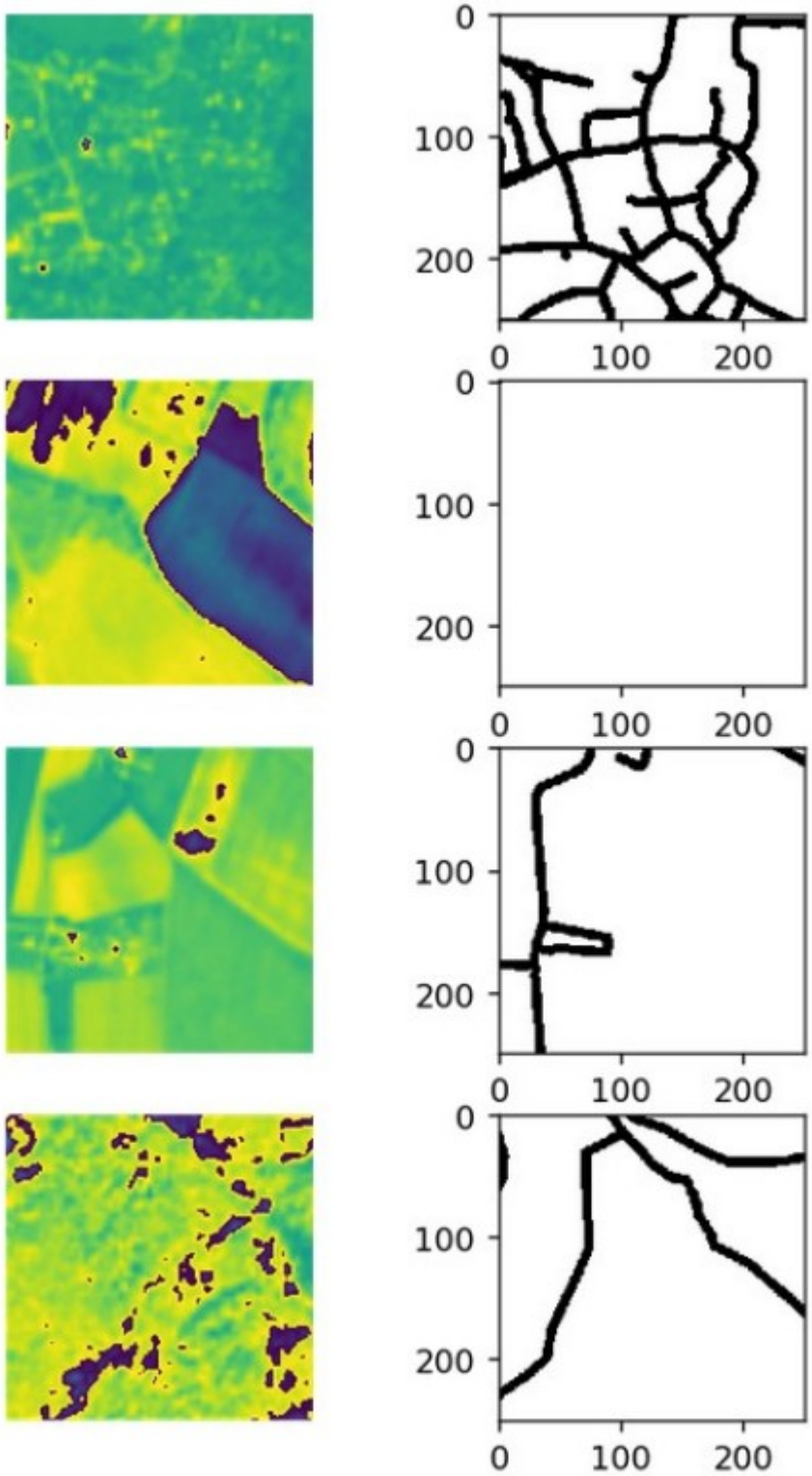
```python
import numpy as np
import sqlalchemy as sa
from wkb_raster import read_wkb_raster
from io import BytesIO

pg = sa.create_engine('postgresql://o:xxx@127.0.0.1:5432/ngi')
sql = "SELECT ST_AsBinary(rast) AS data FROM ngi.rgb LIMIT 10"
lines = pg.execute(sql)

for line in lines:
    read_wkb_raster(BytesIO(line.data))['bands'][0]['ndarray']
```

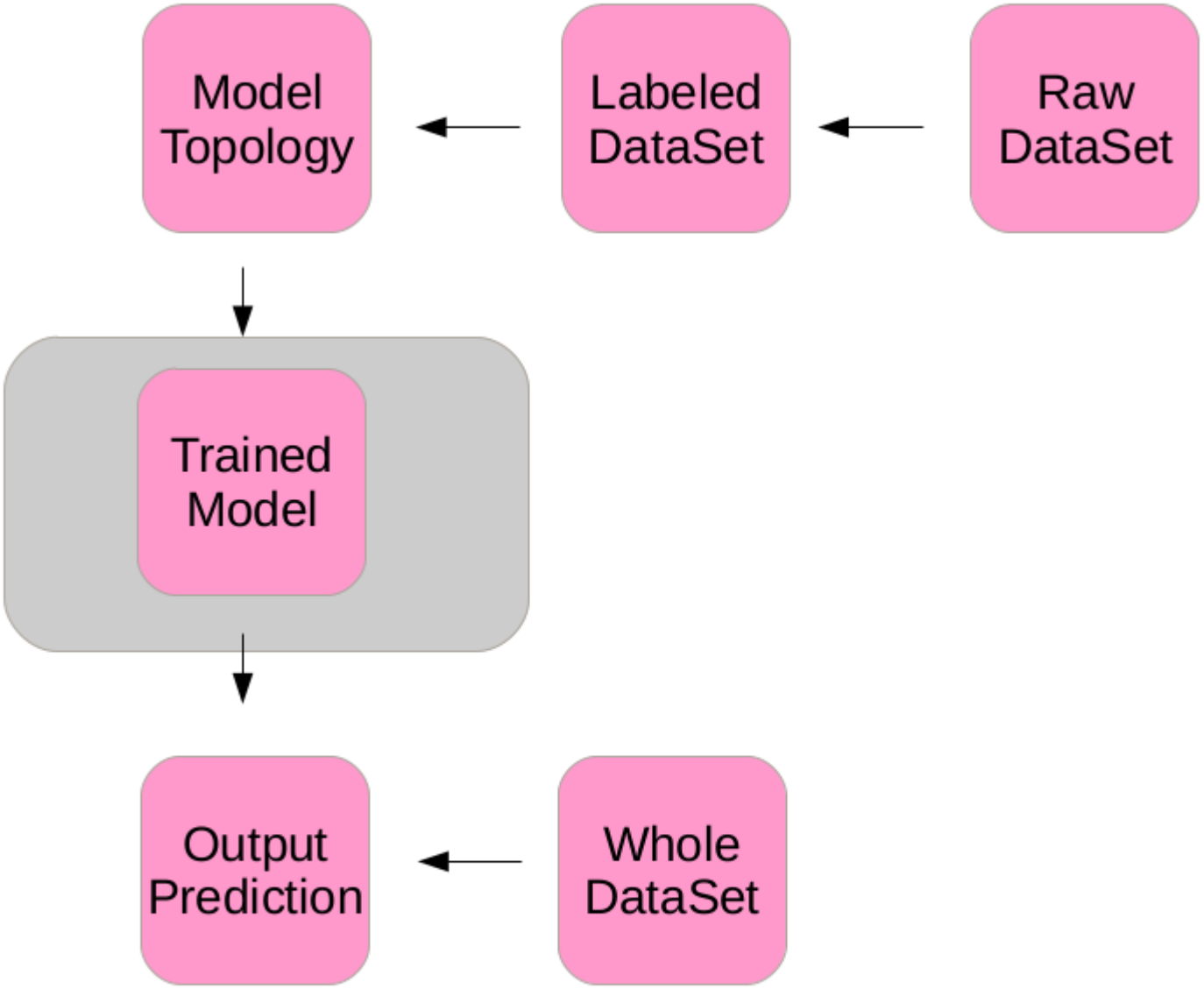
```python
class SimpleIter(mx.io.DataIter):
    def __init__(self, data_names, data_shapes, data_gen,
                 label_names, label_shapes, label_gen, num_batches=10):
        self._provide_data = zip(data_names, data_shapes)
        self._provide_label = zip(label_names, label_shapes)
        self.num_batches = num_batches
        self.data_gen = data_gen
        self.label_gen = label_gen
        self.cur_batch = 0

    def __iter__(self):
        return self

    def reset(self):
        self.cur_batch = 0

    def __next__(self):
        return self.next()

    @property
    def provide_data(self):
        return self._provide_data

    @property
    def provide_label(self):
        return self._provide_label

    def next(self):
        if self.cur_batch < self.num_batches:
            self.cur_batch += 1
            data = [mx.nd.array(g(d[1])) for d,g in zip(self._provide_data, self.data_gen)]
            label = [mx.nd.array(g(d[1])) for d,g in zip(self._provide_label, self.label_gen)]
            return mx.io.DataBatch(data, label)
        else:
            raise StopIteration
```

https://mxnet.incubator.apache.org/tutorials/basic/data.html

```
WITH
origins AS (SELECT ('{{855878,6534055},{878721,6533022},{873294,6541341},{870027,6524893}}'::float[]) AS ul ),
tiles AS (
     SELECT row_number() OVER() as tid,
            ST_SetSRID(
                ST_MakeEnvelope(ul[i][1], ul[i][2], ul[i][1] + 1250, ul[i][2] + 1250)
                , 2154
            ) AS geom
     FROM origins, generate_subscripts((SELECT ul FROM origins), 1) AS i
  ),
tile_rast AS
(
    SELECT  tiles.tid,
            ST_AddBand(
                ST_SetSRID(
                    ST_MakeEmptyRaster(
                        250, 250,
                        ST_Xmin(tiles.geom)::float8,
                        ST_Ymax(tiles.geom)::float8,
                    2.5),
                2154),
            '8BUI') AS rast
    FROM tiles
),
images AS
(
    SELECT  tile_rast.tid,
            tile_rast.rast AS tile_rast,
            ST_MapAlgebra(
                ST_AddBand(tile_rast.rast, '8BUI'::text), 1,
                ST_Resample(ST_Grayscale(ST_Union(image.rast)), tile_rast.rast, 'bilinear'), 1,
                '[rast2] ', NULL, 'FIRST', '[rast2]'
            ) AS rast

    FROM tile_rast, LATERAL
    (
        SELECT rast
        FROM sat.s2
        WHERE ST_Intersects(s2.rast, tile_rast.rast)

    ) AS image

    GROUP BY tile_rast.rast, tile_rast.tid
  ),
labels AS (
    SELECT  tile_rast.tid,
            ST_MapAlgebra(
                 tile_rast.rast,
                 ST_AsRaster(label.geom, tile_rast.rast, '8BUI'),
                 '([rast2])::integer', NULL, 'FIRST', '([rast2])::integer'
             ) AS rast

    FROM tile_rast, LATERAL
    (
       SELECT ST_ClipByBox2D(ST_Buffer(ST_Union(osm.way), 10),
                             ST_Envelope(tile_rast.rast)) geom
       FROM planet_osm_line osm
       WHERE osm.highway IS NOT NULL AND (osm.route = 'road' OR osm.route IS NULL)
       AND ST_Intersects(osm.way, tile_rast.rast)

       GROUP BY tile_rast.rast, tile_rast.tid

    ) AS label
)
SELECT Box3D(images.rast) AS bbox,
       ST_AsBinary(images.rast) AS data,
       CASE WHEN labels.rast IS NULL
           THEN ST_AsBinary(images.tile_rast)
           ELSE ST_AsBinary(labels.rast)
       END AS label
FROM  labels RIGHT JOIN images ON images.tid = labels.tid
```

```
batch_size = 2
max_iter = 2

geo_iter = GeoIter(
    'postgresql://o:xxx@127.0.0.1:5433/osm_qa',
    (850000,6524040,890960,6565000), 2154, (256, 256), (10, 2.5),

    """
         SELECT ST_ClipByBox2D(ST_Buffer(ST_Union(osm.way), 6),
                               ST_Envelope(tile_rast.rast)) geom
         FROM planet_osm_line osm
         WHERE osm.highway IS NOT NULL AND (osm.route = 'road' OR osm.route IS
NULL)
         AND ST_Intersects(osm.way, tile_rast.rast)
    """,

    """
        SELECT ST_Grayscale(ST_Union(s2.rast)) AS rast
        FROM sat.s2
        WHERE ST_Intersects(s2.rast, tile_rast.rast)
    """,

    batch_size, max_iter)
```


0
100
200
0
100
200

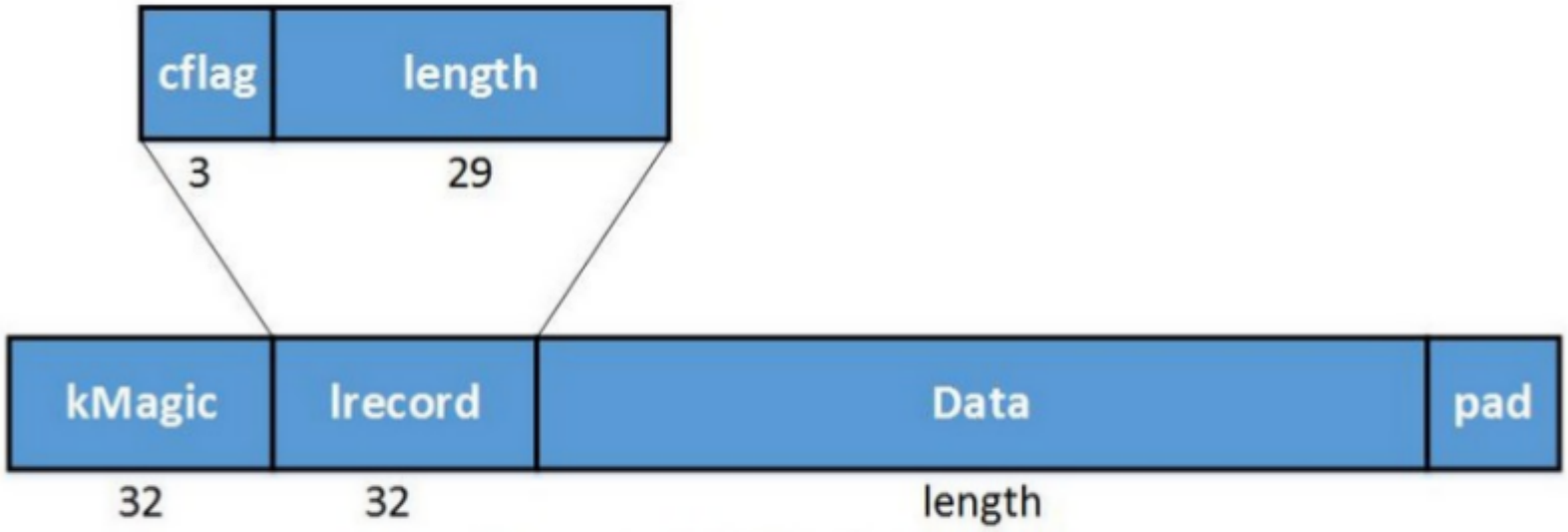

Model Topology
Labeled DataSet
Raw DataSet
Trained Model
Output Prediction
Whole DataSet

MxNet RecordIO fast data loader

Data Format

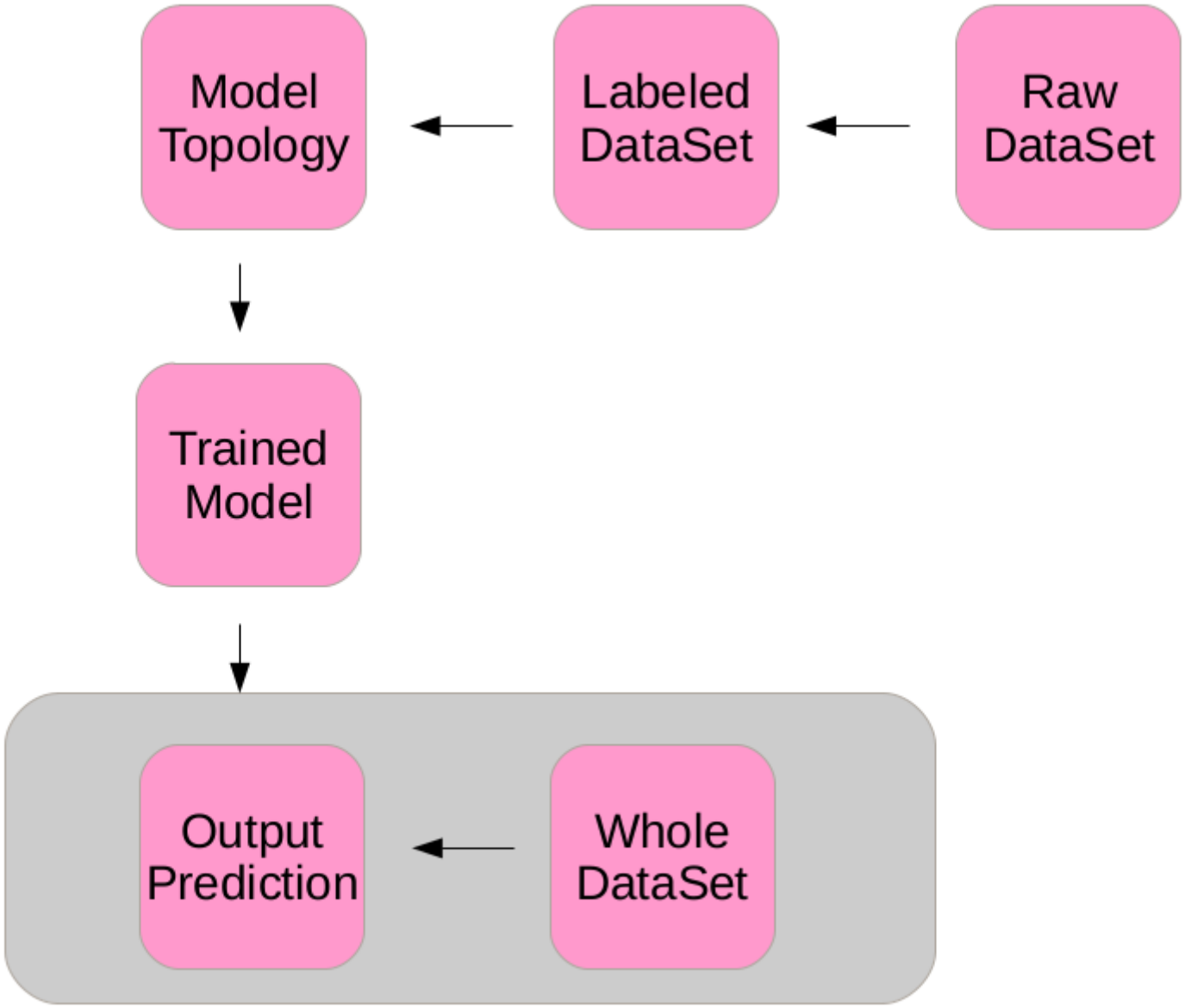
Since the training of deep neural network often involves large amounts of data, the format we choose should be both efficient and convenient. To achieve our goals, we need to pack binary data into a splittable format. In MXNet, we rely on the binary recordIO format implemented in dmlc-core.

Binary Record



Binary recordIO Basic Format

https://mxnet.incubator.apache.org/architecture/note_data_loading.html


Model Topology
Labeled DataSet
Raw DataSet
Trained Model
Output Prediction
Whole DataSet

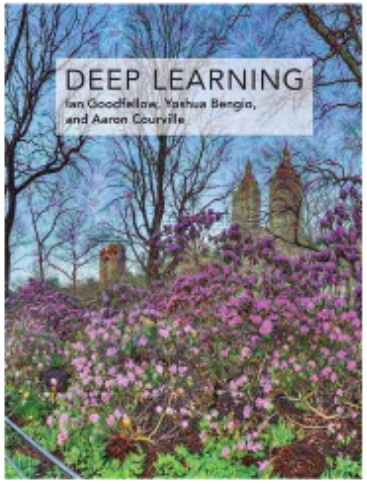
Human Learning

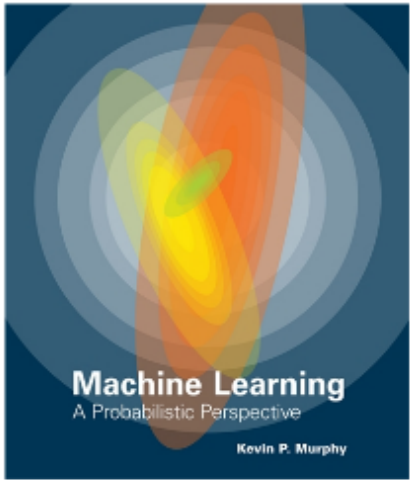
https://www.college-de-france.fr/site/yann-lecun/course-2015-2016.htm

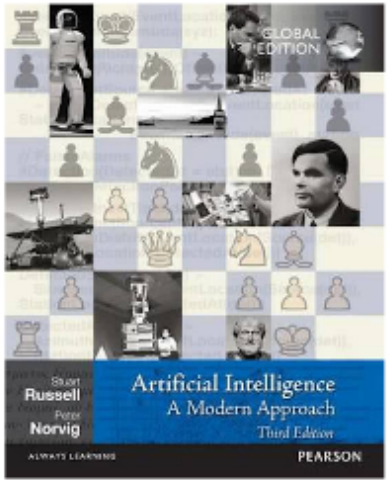
http://cs231n.stanford.edu/syllabus.html

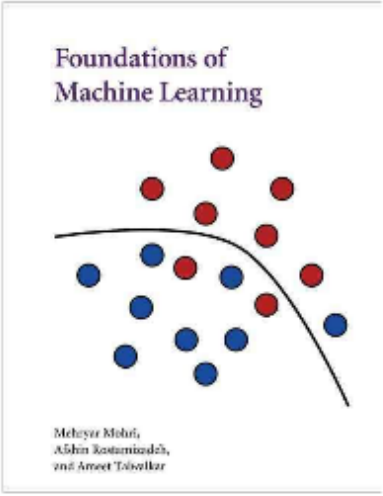
https://raw.githubusercontent.com/mrgloom/Semantic-Segmentation-Evaluation/master/README.md

https://arxiv.org/abs/1802.01528v2









Conclusions

@data_pink

www.datapink.com