EPFL
EXTENSION
SCHOOL

EPFL Extension School Workshop
Machine Learning Part

AMLD 2020

Plan for this morning

- Who are we?

- Hands-on experience through 4 Machine Learning use cases
  - Anomaly Detection
  - Text classification
  - Image Classification
  - Face Recognition

- Q&A

~30min Coffee Break in the middle!

A few words about us

EPFL Extension School

Online self-paced learning platform, COS from EPFL, ~450 hours (15 ECTS)

Applied Data Science: Machine Learning Program

Bring professionals from different backgrounds to acquire Machine Learning skills

FRED OUWEHAND
Senior Course Developer and Instructor

PANAGIOTA XYDI
Course Developer and Instructor

MICHAEL NOTTER
Course Developer and Instructor

CHRISTIAN LUEBBE
Course Developer and Instructor

What do we cover in our courses?

1. Intro to Data Analysis with Python

- Foundational concepts of data analysis
- Python Recap
- Data handling & data visualization
- Course Project 1

2. Applied Data Analysis

- Getting, cleaning & manipulating data
- Working with different data types (text, images, time series, ...)
- Statistical data analysis & data exploration
- Course Project 2

What do we cover in our courses?

3. Applied Machine Learning (Basics)

- Introduction to ML concepts: Feature engineering, model optimisation & evaluation, reliability of results
- First steps with scikit-learn (major ML toolbox in Python)
- Linear and polynomial regression
- Course Project 3

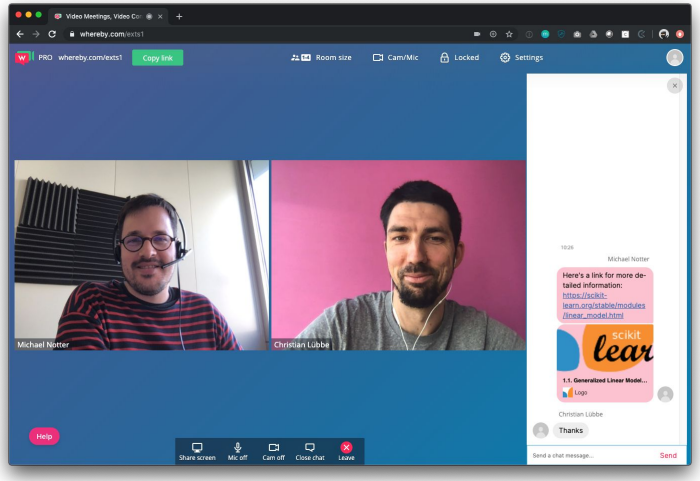
4. Applied Machine Learning (Advanced)

- Practical experience with classification, regression, clustering
- Advanced ML models: KNN, Logistic Regression, Decision Trees, SVM, K-means
- Deep Learning with Tensorflow
- Course Project 4

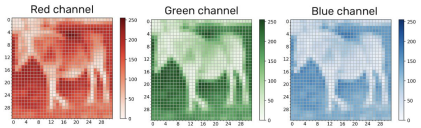
5. Customizable Capstone Project

We offer personalized support!

1-1 video call
with experts for ~1h



This code is also a nice way to visualize the 3-dimensional `img` array. You can think of it as three 32 by 32 grids stacked on top of each other. In this visualization, `img[16, 1, 2]` denotes the pixel at (16, 1) in the third grid which corresponds to the horse nostrils in the blue channel.



Summary

In this unit, we saw how to load images into Numpy arrays using the `open()` function from PIL, and how to plot them using the `imshow()` one from Pyplot. In particular, we saw that

- Grayscale images correspond to 2-dimensional `(height, width)` arrays of white intensity values
- Color images correspond to 3-dimensional `(height, width, channels)` arrays
- Pixel values vary between 0 and 255 and we store them with the `uint8` data type

In the next unit, we will see how to convert these arrays of pixels into a set of features that we can use in our machine learning algorithms.

QUESTIONS — Ask a Question

· Learner · 4 months ago · Edit · Hide — ✓ 2 Answers

high definition & color picture reading

hello is there also a way to import high definition (x Mio pixel) color picture(16/24/36/48bits)(jpg, ...) , how will they be represented? thanks

Frederic Ouwehand · Teacher · 4 months ago · Time spent: 00:20 · Edit · Convert to question — 0

Hello. I would still use `img = Image.open()` , convert the PIL image into Numpy format with `np.array(img)` and then work on subset of the image ex. `Image.fromarray(img[:128, :128])` , but it's true that those specific formats doesn't seem to be supported by Pillow: list of modes - did you try with OpenCV?

· Learner · 4 months ago · Edit · Convert to question — 1

hello frederic, no i didnt try it, lol, but i will look for... thank you very much

In-platform support
from instructors

COMMENTS — Post a Comment

Michael Notter · Teacher · 09-01-2020 20:44 · Time spent: 00:20 · Edit · Destroy

Hello ,

Thank you for your project solution. You completed all the tasks and did excellent work throughout. Well done!

The following is a list of comments and thoughts I had while reviewing your solution:

- Overall, your notebooks are very well done! They are well structured, well commented and the code is well written, bravo!
- In the 2nd task of the warm up exercise, you could have also done the `z_score` computation in one go with something like `X_tr = X_tr.mean(axis=0) / X_tr.std(axis=0)` . Or even more compacter, the whole outlier detection with `np.any(np.abs(X_tr - X_tr.mean(axis=0) / X_tr.std(axis=0))>2, axis=1)` . Your solution is correct as well, however `for` loops like this become quickly demanding in respect of computation, which might slow this step down if it is done in pure Python. My compact solution from before uses built in function from `numpy` , which in the backend all use much faster C++ code to compute.
- Your filling missing values section in the house price solution is very well done! Thank you for commenting so well your thoughts. As additional notes, filling up numerical (i.e. continuous) values with the mean is often a good idea, but keep in mind that the mean can be influenced by outliers or skewed distributions. For this reason I recommend to use the median in such a case. Additionally, if you want to spend extra energy on it, you could also fill up missing values through a more nuanced strategy. For example, fill them up with the mean, median or mode value from houses in the same neighbourhood, or of the same size etc.
- In the section "Look for inconsistencies values", you plot the individual values with a `plot()` , in such a situation where the consecutive values have no relationship, I would rather recommend to use a scatter plot. Ideally also reduce the size of the dots in the scatter plot to better observe the nuanced value distribution.
- You also mention to put them in relation with `SalePrice` . That's a great idea! I would recommend to use something like `sns.jointplot()` to have a lot of information at once:

```
import seaborn as sns
for col in data.select_dtypes(np.number).keys():
    sns.jointplot(data['SalePrice'], col, data=data, height=6, s=1)
plt.show()
```

- Your data cleaning in the house prices exercise is great! It's very well structured and easy to understand what you thought and did. You engineered also many new very interesting features.
- Instead of `correlation[(correlation > 0.6) | (correlation < -0.6)]` , you could have also written `correlation[np.abs(correlation) > 0.6)]`
- You've written two functions `func` and `bool_val` to handle boolean values [illegible]em. This is not necessary for the data modelling. Scikit-learn is aware about the boole[illegible]orms this on the fly to 0s and 1s if necessary. So, if you have a variable `x` with a[illegible] do `~x` to invert all its values.
- It's great that you put all preprocessing and clea[illegible]pply it to the test set as well. But keep in mind, that any[illegible] the test set needs to be identical to the one a[illegible]es with the `mean` or `mode` , then you[illegible]wise you might introduce an[illegible]

The perform[illegible]! And running the model with with c[illegible]and important data science concepts su[illegible]

I'm also happy [illegible]withheld test set was very good. You've reached the following MAE:

- simple: 18'9[illegible]
- intermediate: [illegible]
- complex: 12'692[illegible]

This values are not just very good, they are also very close to the ones that you have on the validation set. Well done as well!

You can now proceed with the next course. Don't hesitate to book a 1-1 video call with me if you have any questions about the course or want to discuss the project results in more details. You can do it directly via this link.

Best regards,
Michael

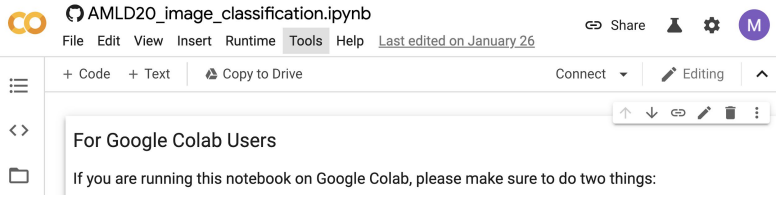
Detailed Project Feedback

Hands-on experience through 4 Machine Learning use cases

1. Anomaly Detection
2. Text classification
3. Image Classification
4. Face Recognition

Quick Tools Introduction

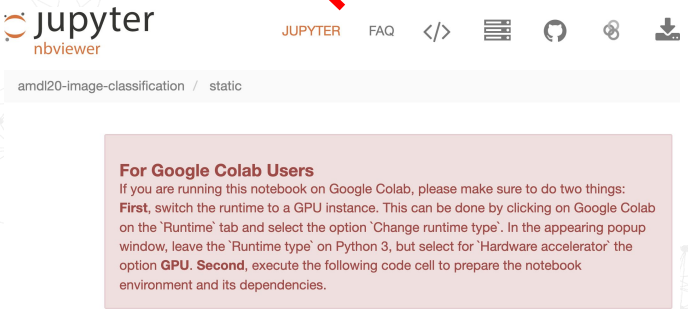
Open in Colab | launch binder | Offline View Open



Google Login Required
(Provides powerful machines)

Starting repository: epfl-exts/amdl20-image-classification/master

No Login Required
(Provides weak machines)



No Login Required
(Static view of final output)

Anomaly Detection

What are anomalies?

Other names: Outlier, novelty, unusual events

Definition: *“An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism”* D.W. Hawkins

Problem:

1. Anomalies are rare or even unseen events
2. There is not template to detect them
3. Can be costly if undetected

Applications of anomaly detection

1. Credit card fraud
2. Network intrusions / cyber attacks
3. Failure of sensors or machines (predictive maintenance)
4. Defective products in a production cycle
5. Astronomy

GitHub repository

github.com/epfl-exts/amld20-anomaly-detection

Demonstration

Data: Network data (adapted from NSL-KDD data)

Task: Identify attacks (anomalies)

Steps:

1. Create your training data
2. Explore attack distribution in training data
3. Build anomaly detector using an Isolation Forest
4. Predict on new data
5. Evaluate success

“20 Questions” meets “Top Trumps”

Game rules:

- Cards with technical data of cars: *length, weight, HP, engine size, year built, ...*
- You pick a car
- I ask Yes/No questions until I have identified your car

Which cars could I identify quickly?
They are unusual in some way.

Isolation Forest algorithm

Start with the full pile of cards (dataset)

1. Pick a pile with more than 1 card
2. Randomly select a feature and a split value
3. Divide the pile into two piles

Isolation Forest algorithm

Start with the full pile of cards (dataset)

1. Pick a pile with more than 1 card
2. Randomly select a feature and a split value
3. Divide the pile into two piles
4. Repeat steps 1-3 until each pile has only one card (i.e. the card is isolated)
5. For each card record number of splits needed to isolate it

Isolation Forest algorithm

Start with the full pile of cards (dataset)

1. Pick a pile with more than 1 card
2. Randomly select a feature and a split value
3. Divide the pile into two piles
4. Repeat steps 1-3 until each pile has only one card (i.e. the card is isolated)
5. For each card record number of splits needed to isolate it
6. Repeat steps 1-5 several times and average number of splits needed
7. Few splits => easy to isolate => more likely anomalous
8. Rank by average number of splits

Hands-on

Your tasks:

Explore the notebook

Change the different parameters and see how the outcome varies

Text classification

Applications of text classification

1) Spam filters

2) Social media monitoring (e.g. automatic hate-speech detection, abusive language, ...)

3) Sentiment Analysis or opinion mining

4) News articles categorization

5) ...

GitHub repository

github.com/epfl-exts/amld20-text-classification

Text preprocessing

HTML tags

<html> <body bgcolor="#FFFBE0"> smart shoppers click here for the best rates
 <table border="0" cellpadding="2" cellspacing="0"> <tr> <td bgcolor="#000000"> <table bgcolor="#ffffff" border="0" width="470" cellpadding="8" cellspacing="0"> <tr> <td align="center"> <font face="arial" size="5" color="#8B0909"><b>Paying Too Much for Life Insurance?
<a href="http://www.marketing-leader.com/user0202/instant_savings"><font face="verdana" size="4" color="#0000ff"> Click Here to Save 70% on Your Policy...

Text preprocessing

URLs

Do you want to make money from home? Are you tired of working for someone else? Then welcome to the future. **http://www.lotsonet.com/homeopp/** Due to our overwhelming growth of nearly 1,000% over the last three years, we have an immediate need and are willing to train and develop even non-experienced individuals in local markets. **http://www.lotsonet.com/homeopp/**

E-mail addresses

Website: http://www.nmtbmedia.com - ---------- >-----Original Message----- >From: **ilug-admin@linux.ie** [**mailto:ilug-admin@linux.ie**]On Behalf Of >Brian O'Donoghue >Sent: 16 August 2002 17:42 >To: '**ilug@linux.ie**' >Subject: RE: [ILUG] ADSL, routers and firewalls > > >> To get the 3 computers talking to each other, you need a hub >which you can >> pick up for a small amount of money...

Text preprocessing

Punctuation marks, special characters, digits, multiple whitespace...

Website: http://www.nmtbmedia.com - ---------- >-----Original Message----- >From: ilug-admin@linux.ie [mailto:ilug-admin@linux.ie] On Behalf Of >Brian O'Donoghue >Sent: 16 August 2002 17:42 >To: 'ilug@linux.ie' >Subject: RE: [ILUG] ADSL, routers and firewalls > > >> To get the 3 computers talking to each other, you need a hub >which you can >> pick up for a small amount of money. The internal interface of >the router >> (which may be a linux box) is connected to the hub, and the >> external interface is connected to the ADSL device. > > >Like this > > >[DSL Connection]--------[Alcatel modem]---[ehternet card1]-[Linux >box] >[ ] > [Hub]----------[Ethernet card2]-[>] > | > | > | > [Other machines on...

Github Repo: github.com/epfl-exts/amld20-text-classification

Text preprocessing

Remove stopwords (“a”, “the”, “and”, “by”, “about”, “his”…)

Example:

“We are amazing!”

Case conversion

Example:

“Dog” and “dog”

Examples of “cleaned” text

Original text

URL: http://jeremy.zawodny.com/blog/archives/000212.html Date: 2002-10-02T09:34:00-08:00 Talking about his mainframe background and roots in computing. He played a lot on the big iron and sorta "missed" the PC revolution. He wasn't terribly interested in PCs for quite a while. Then he got to play with one...

Cleaned text

date talking mainframe background roots computing played iron sorta missed revolution wasnt terribly interested quite play

Examples of “cleaned” text

Original text

HI, zzzz@spamassassin.taint.org today, <html> <head> <title>Ink Price</title> <meta http-equiv="Content-Type" content="text/html;"> </head> <body bgcolor="#ffffff"> </td> </tr> <tr> <td><img name="smallad1_r2_c1" src="http://www.%43%4f%6d%6f-%56e%6e%44%45%52.com/ink/ad/smallad1_r2_c1.jpg" width="365" height="86" border="0"></td> </tr> </table> _____ If you would not like to get more spacial offers from us, please CLICK HERE and you request will be honored immediately!
 _____   </body> </html>

Cleaned text

today price like spacial offers click request honored immediately

Error analysis

Original text

Dear Mr Mason, With Christmas just round the corner, here's some offers worth celebrating. If you want to start stocking up on beer and spirits for the festive season, then we want to make it better value for money for you. Plus we've some great offers on frozen turkeys and mince pies. Cost Price Guinness and Budweiser ---------- Budweiser 500ml can x 24 Now 42.14 Euro Save 5.62 Euro Guinness Draught 500ml can x 24 Now 42.32 Euro Save 5.44 Euro Offers available until 04/12/02, while stocks last* http://www.tesco.ie/Register/ Cost Price Selected Spirits ---------- Jameson Whiskey 1ltr and 70cl Cork Dry Gin 1ltr and 70cl Smirnoff Red Label Vodka 1ltr and 70cl Hennessy Cognac 1ltr and 70cl Baileys Original Irish Cream 1ltr and 70cl Bacardi White Rum 1ltr and 70cl Offers available until 04/12/02, while stocks last* http://www.tesco.ie/Register/ 50% Off Frozen Turkeys ---------- This great offer is available on Tesco's Frozen Turkey range from 3.5kg to 9.5kg in weight Offers available until 01/12/02, while stocks last http://www.tesco.ie/Register/ 20% Off Mince Pies ---------- Sweetie Pies! Take advantage of this tempting offer Offers available until 01/12/02, while stocks last http://www.tesco.ie/Register/ Tesco.ie Christmas Zone ---------- Visit our Christmas zone online at www.tesco.ie

Error analysis

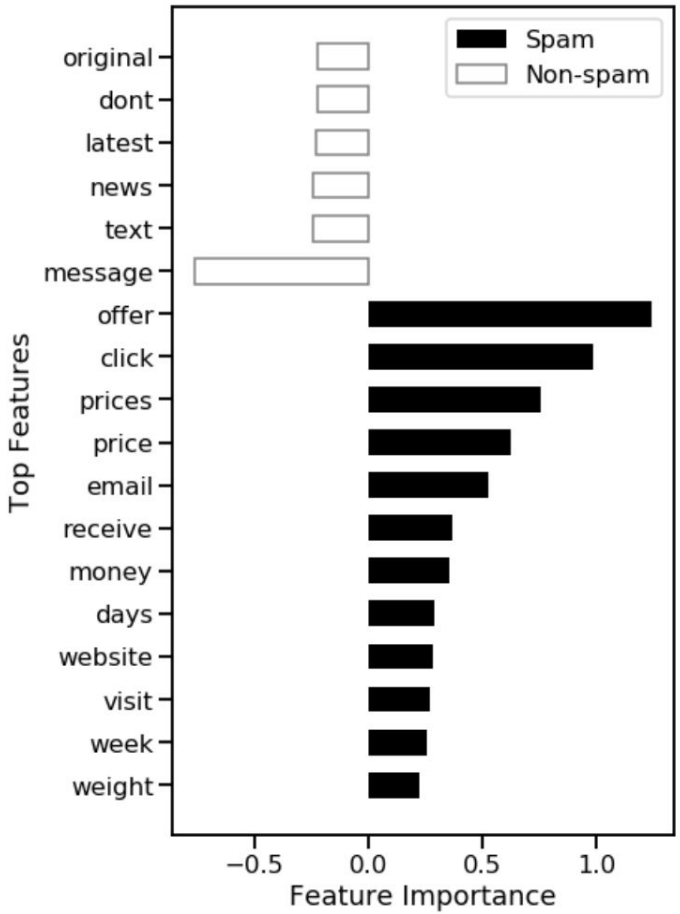
Actual class: Non-Spam

Predicted class: Spam

Predicted probabilities

Non-spam: 0.1%

Spam: 99%



Github Repo: github.com/epfl-exts/amld20-text-classification

Error analysis

Original text

µÚÊ®Ò»½ìÖÐ¹ú¹ú¼Êµç×ÓÉè±¸¡¢µç×ÓÔªÆ÷¼þ¡¢±íÃæÌù×°Õ¹ÀÀ»á½«ÓÚ2002Äê4ÔÂ1ÈÕ--4ÈÕÔÚÉîÛÚÖÐ¹ú¹ú¼Ê¸ßÐÂ¼¼Êõ³É¹û½»Ò×»áÕ¹ÀÀÖÐÐÄÂ¡ÖØ¾ÙÐÐ¡£ ÓûÖªÕ¹»áÏêÇé£¬¾´ÇëµÇÂ¼ÎÒÃÇÍøÕ¾£ºhttp://www.e-dowell.com£¨¶àÈËÐÐµç×Ó½»Ò×Íø£©¡£
ÔÚÉç»á¸÷½çµÄ´óÁ¦Ö§³ÖÏÂ£¬¶àÈËÐÐµç×ÓÕ¹ÒÑ¾¾É¹¾Ù°ìÁËÊ®½ì£¬¸ÃÕ¹»áÒò²Î¹ÛÕßÖÚ¶à¡¢¶©»õÁ¿´ó¶øÉî»ñ¹ã´ó³§ÉÌºÃÆÀ¡£¸ÃÕ¹ÒÑ³ÉÎªÒ»Äêò»¶ÈµÄµç×ÓÐÐÒµÊ¢»á£¬ÊÇ¹úÄÚÍ¬ÐÐÒµÖÐ×î¾ßÓ°ÌìÁ¡¡¢ºÅÕÙÁ¡µÄ´óÐÍÕ¹ÀÀ»áÖ®Ò»¡£ÎÒÃÇ½«ÓÚ2003Äê4ÔÂ1ÈÕÖÁ4ÈÕ¾Ù°ìµÚÊ®Ò»½ìÖÐ¹ú¹ú¼Êµç×ÓÉè±¸¡¢µç×ÓÔªÆ÷¼þ¡¢±íÃæÌù×°Õ¹ÀÀ»á¡£ Ê±Öµ¹úÇì½Ú½«ÖÁ£¬¶àÈËÐÐ¹«Ë¾×£Äú½ÚÈÕÓä¿ì£¬ÊÂÒµÓÐ³É£¡2003 the 11th China International Electronic Equipments and Components & Surface Mounting Technology Fair ÐÅÏ¢²úÒµ²¿¡¢¿ÆÑ§¼¼Êõ²¿¡¢ÖÐ¹úµç×ÓÑ§»á¡¢ÉîÛÚÊÐÕþ¸®ÖØµãÖ§³ÖµÄ´óÐÍµç×Ó×¨ÒµÕ¹Ê±¼ä£º
2003Äê4ÔÂ1ÈÕÖÁ4ÈÕ TIME£ºApril 1st to 4th , 2003 µØµã£º ÉîÛÚÖÐ¹ú¹ú¼Ê¸ßÐÂ¼¼Êõ³É¹û½»Ò×»áÕ¹ÀÀÖÐÐÄ ADD£ºCHINA HI-TECH FAIR EXHIBITION CENTRE £¨SHENZHEN£© Åú×¼µ¥Î»: ÖÐ»ªÈËÃñ¹²Í¹ú¿ÆÑ§¼¼Êõ²¿ Approver: Ministry of Science and Technology of People's Republic of China Ö÷°ìµ¥Î»/Sponsors: ÖÐ¹úµç×ÓÑ§»á Chinese Institute of Electronics (CIE) Ð°ìµ¥Î»/CO-ORGANIZER£º ÖÐ¹ú°ëµ¼ÌåÐÐÒµÐ»á China Semiconductor Association ÖÐ¹úµç×Ó±¨Éç China Electronics News Agency ³Ð°ìµ¥Î»/Organizer £º ÖÐ¹úµç×ÓÑ§»áÕ¹ÀÀ²¿ Exhibition Dept. of Chinese Institute of Electronics ÉîÛÚÊÐ¶àÈËÐÐÊµÒµÓÐÏÞ¹«Ë¼ SHENZHEN DOWELL INDUSTRIA

Error analysis

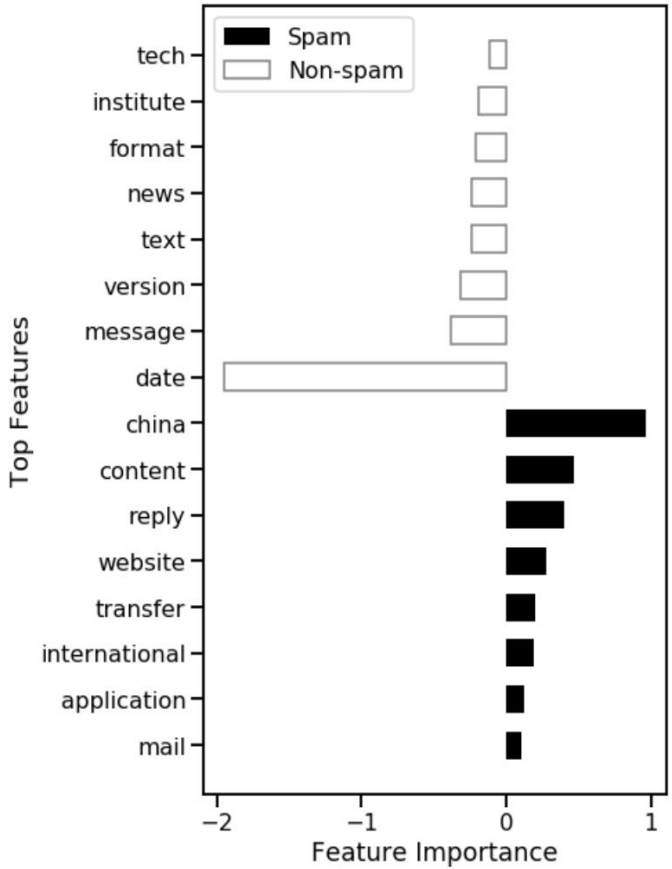
Actual class: Spam

Predicted class: Non-spam

Predicted probabilities

Non-spam: 60%

Spam: 40%



Github Repo: github.com/epfl-exts/amld20-text-classification

Coffee Break

(till 11:00)

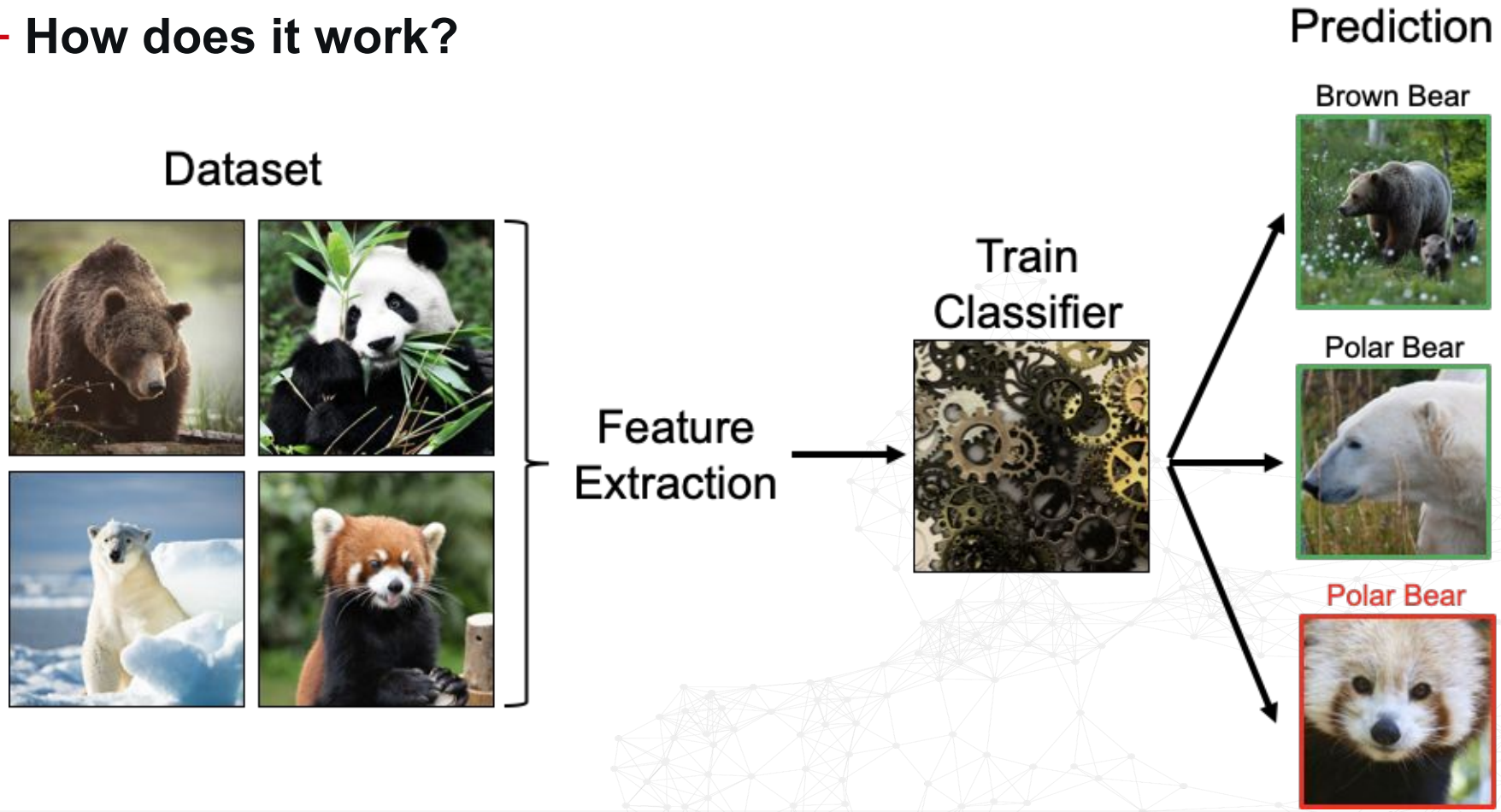
Join Our Team:
https://epflextensionschool.recruitee.com/

Image Classification

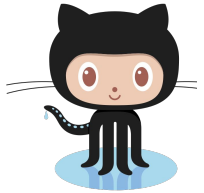
Applications of Image Classification

1. **Automobile Industry**, e.g. self-driving cars
2. **Healthcare Industry**, e.g. classification of X-ray images
3. **Manufacturing Industry**, e.g. detection of faulty outcome in production line
4. ...

How does it work?



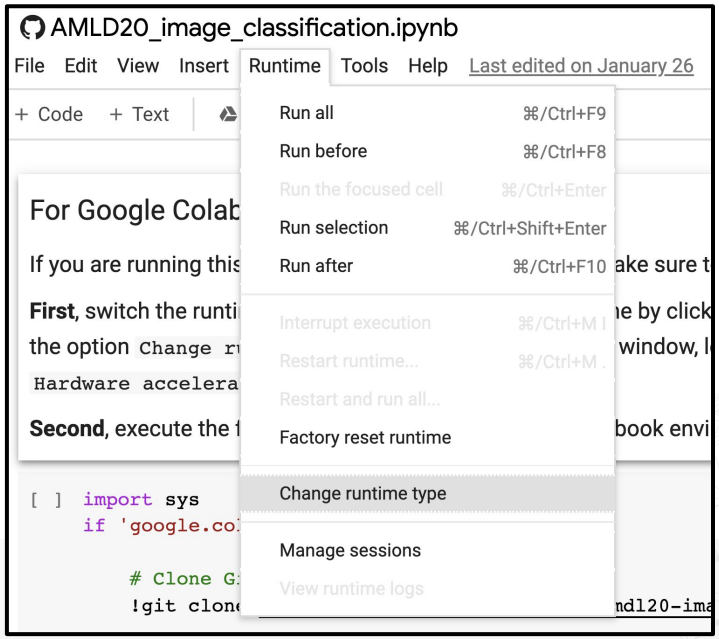
GitHub repository



github.com/epfl-exts/amld20-image-classification

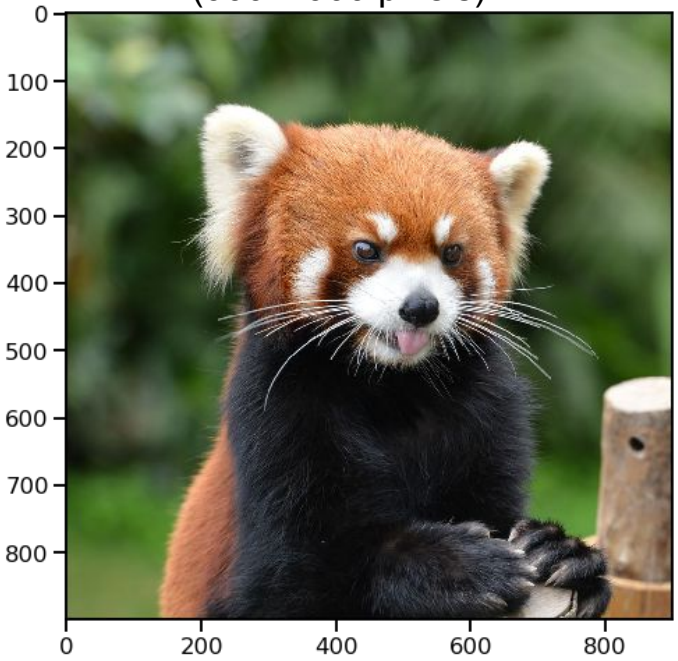
For the content on Colab - switch the runtime to GPU:

Runtime > Change runtime type > Hardware accelerator = GPU



The Data

Original Image
(900 x 900 pixels)



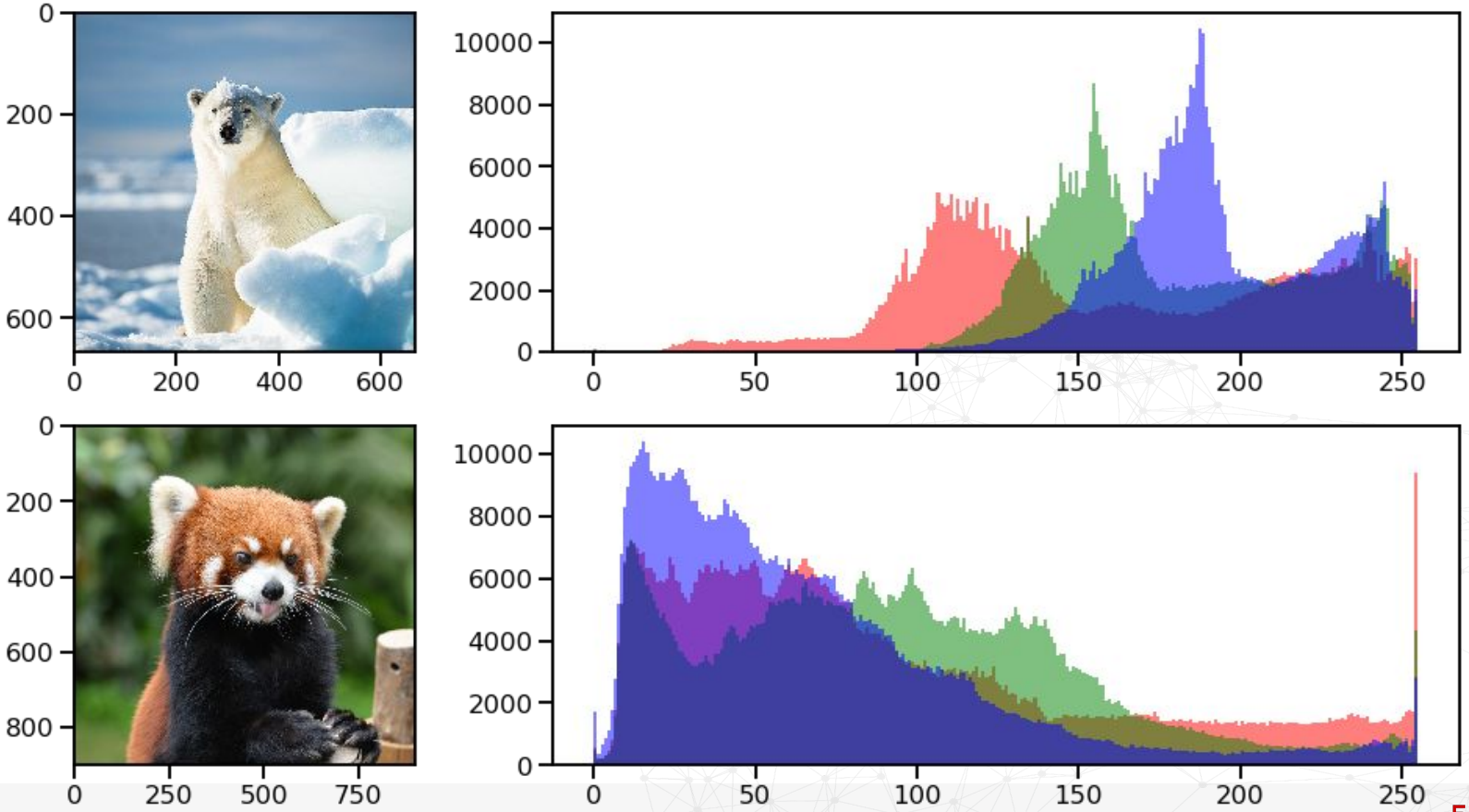
RGB color channels

Cutout of red color channel (30 x 30 pixels)

189	203	189	191	192	192	179	184	197	198	199	207	193	182	172	171	185	195	214	234	237	232	245	248	242	244	225	196	200	168
189	188	206	201	203	222	208	221	220	210	214	202	198	197	181	165	181	181	209	229	236	242	240	246	242	245	219	233	185	174
183	175	194	189	216	204	230	200	221	220	205	190	197	183	178	169	163	173	199	228	234	240	237	244	238	234	215	222	195	200
190	197	202	188	186	200	199	230	227	223	232	209	201	194	184	168	174	155	168	215	222	225	235	238	231	216	211	213	212	176
179	177	185	204	216	218	240	237	232	237	235	236	225	198	192	182	168	148	148	195	213	214	226	226	223	199	211	223	198	165
163	173	177	185	222	231	218	224	223	229	230	232	217	215	209	188	178	159	150	172	199	202	205	205	213	209	214	190	166	159
181	185	197	197	211	212	219	211	207	189	180	198	189	193	197	186	177	166	152	148	182	188	180	181	202	200	185	166	143	132
151	154	155	162	201	201	175	173	144	128	116	106	112	117	132	151	138	159	149	136	149	154	156	157	179	189	158	130	123	113
137	146	164	162	168	174	154	108	75	76	93	69	27	39	56	71	95	116	134	142	121	115	111	127	143	150	114	115	102	95
147	177	177	171	157	141	126	81	85	169	158	153	127	111	77	38	32	53	79	102	94	96	95	112	125	112	99	95	97	90
147	139	149	142	139	136	117	80	130	150	148	160	151	136	138	93	22	25	40	56	73	87	101	109	117	97	94	104	99	98
110	112	112	122	134	122	127	113	120	97	98	108	154	148	118	79	60	24	30	39	52	73	96	109	110	95	106	108	117	130
122	121	129	128	135	136	140	125	123	61	74	75	106	92	66	55	64	38	18	32	48	82	85	102	101	105	98	134	129	153
120	117	109	131	145	133	135	126	114	55	51	65	76	64	52	59	57	49	14	28	57	94	79	96	103	113	122	144	158	173
106	119	112	127	124	124	127	111	105	76	39	60	68	71	52	61	72	31	11	24	54	80	89	114	132	131	156	169	195	193
119	96	96	116	130	123	119	131	97	94	60	40	54	73	72	76	37	27	45	26	63	82	96	122	136	173	188	196	212	197
118	126	103	126	128	128	120	98	109	71	66	58	42	29	38	21	12	9	18	32	61	89	119	133	172	185	207	211	225	218
95	100	124	120	132	133	116	113	111	87	80	57	73	70	78	93	72	69	61	59	63	93	129	146	191	193	215	214	224	227
96	108	94	111	136	120	121	117	103	86	98	86	54	68	72	60	51	60	87	88	94	112	128	166	200	202	217	212	220	223
89	83	124	125	114	109	121	123	133	103	83	86	76	73	70	80	94	127	141	125	143	169	170	196	207	221	224	224	232	228
102	109	77	93	109	110	128	128	132	117	129	95	87	92	96	98	107	122	129	137	171	189	189	199	218	233	232	235	241	234
98	99	98	86	109	120	112	132	118	115	101	117	124	116	123	106	124	134	131	144	184	206	225	226	239	233	237	237	240	239
85	83	78	82	81	107	102	92	111	110	106	109	107	121	121	122	122	127	158	191	234	236	245	241	230	235	243	238	236	242
99	97	92	96	92	76	105	100	73	98	89	86	92	126	114	113	142	153	195	234	239	239	244	248	244	242	244	246	245	241
66	76	80	84	97	85	103	95	102	142	100	116	132	110	118	125	161	204	227	246	247	242	244	247	247	247	247	247	247	244
66	87	95	71	74	109	80	95	102	88	147	178	150	159	163	155	196	222	230	239	245	246	246	246	247	250	248	247	246	244
71	72	92	106	89	88	120	135	138	143	106	126	205	206	221	238	244	242	245	250	255	255	254	251	248	251	248	246	246	246
42	56	72	72	102	122	131	133	145	154	187	204	193	226	225	241	238	249	250	250	250	252	253	250	246	252	250	248	249	248
85	88	82	88	70	111	150	160	188	164	168	204	231	236	242	245	255	248	251	253	251	252	254	252	252	251	250	250	250	249
73	89	103	127	142	111	125	167	187	215	203	209	209	240	248	246	249	252	254	254	252	250	248	249	248	251	251	251	251	248

Github Repo: github.com/epfl-exts/amld20-image-classification

Feature Extraction



Github Repo: github.com/epfl-exts/amld20-image-classification

Demonstration

Data: Collected Images via Google Images

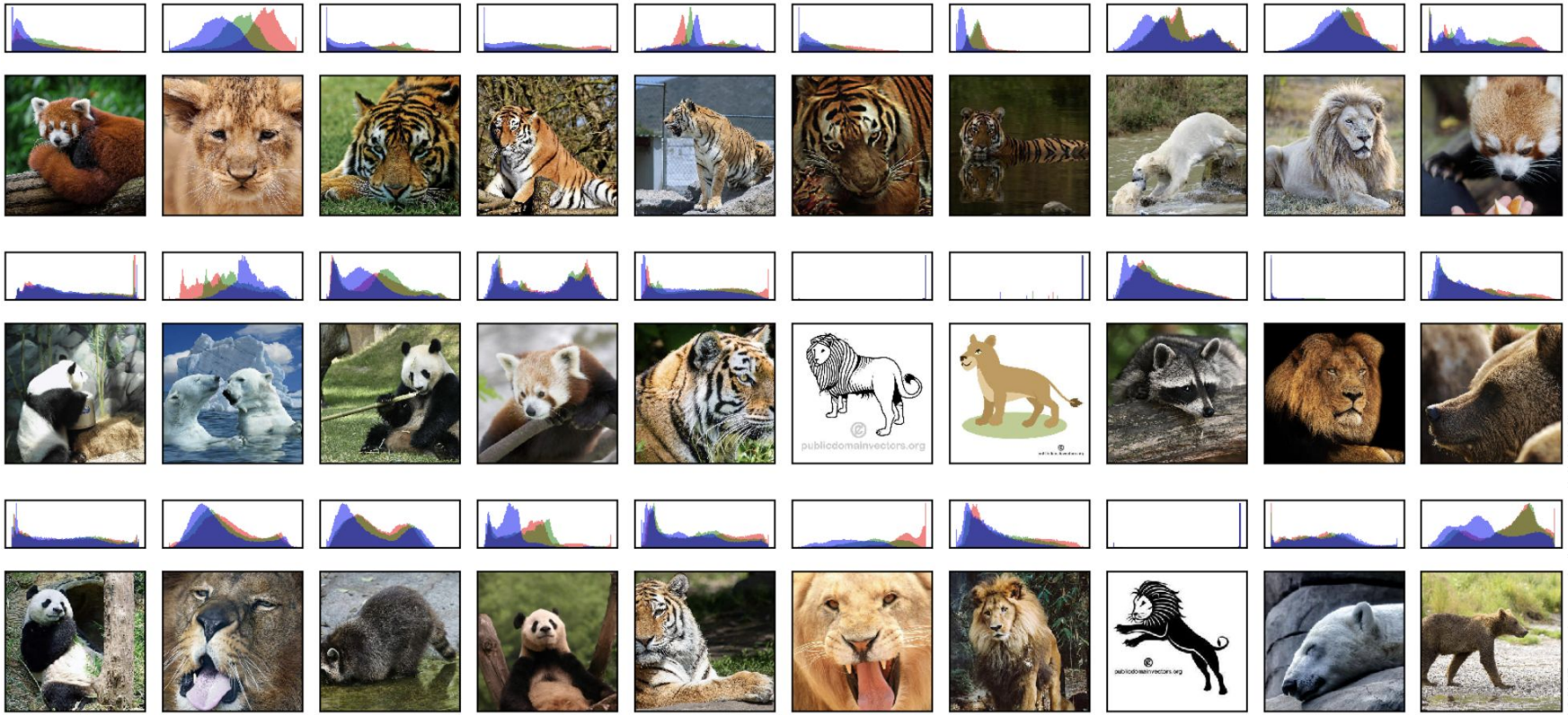
Task: Classify Images

Steps:

1. Data Preparation
2. Data Exploration
3. Data Modeling and Analysis
4. Results Discussion & Exploration

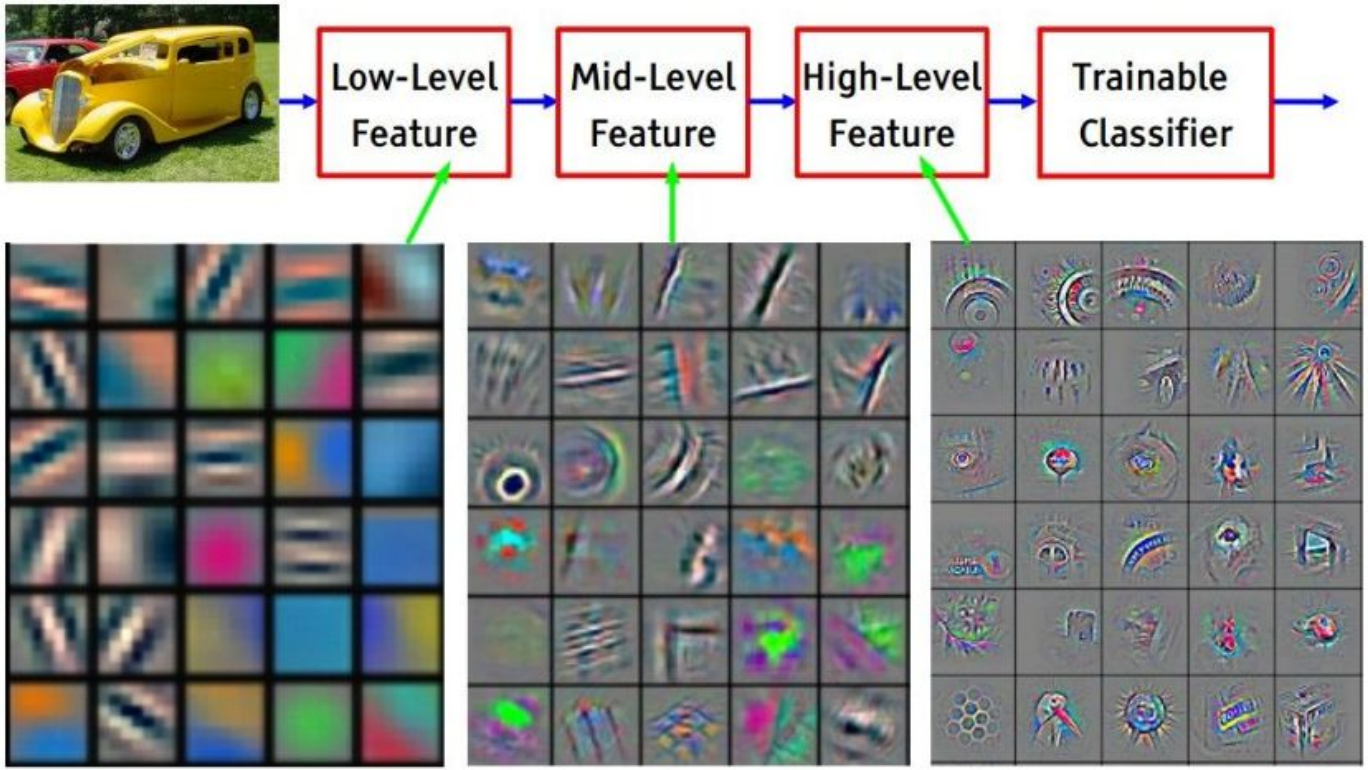
Github Repo: github.com/epfl-exts/amld20-image-classification

Data Preparation: Outlier Detection



Github Repo: github.com/epfl-exts/amld20-image-classification

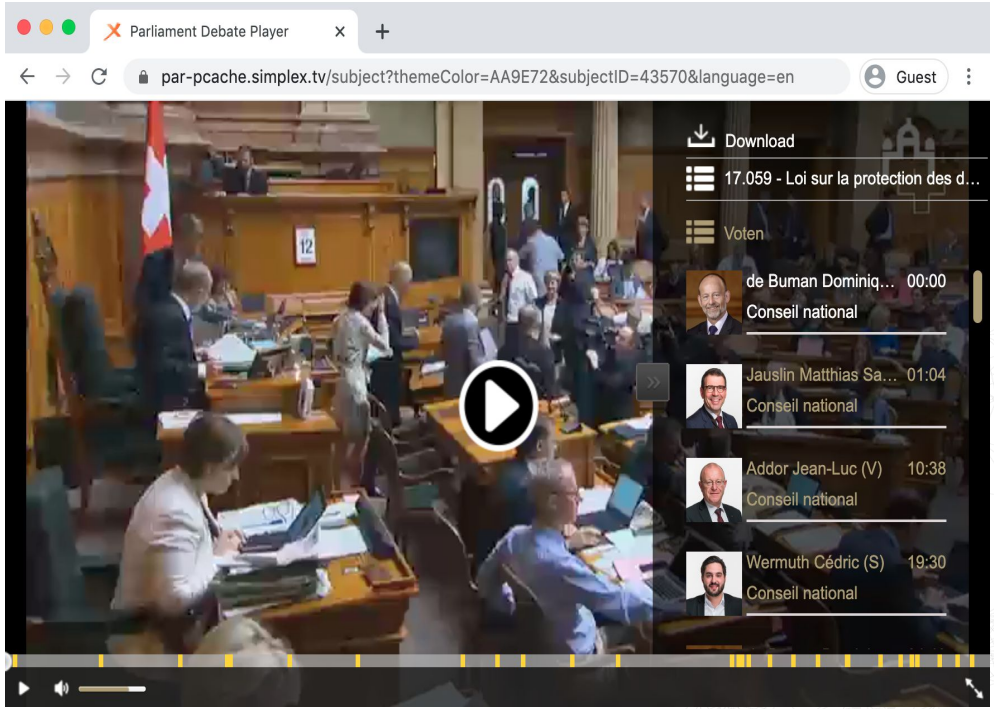
Feature Extraction with Convolutional Neural Network



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013].

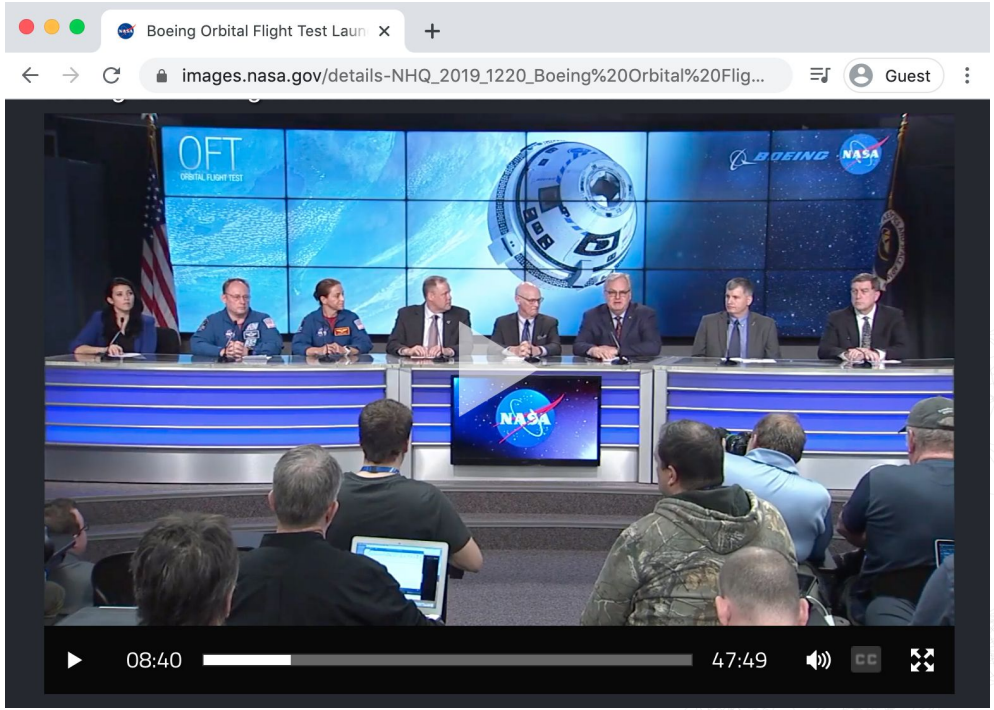
Face Recognition

Video search - Parliament debates



Source parliament.ch

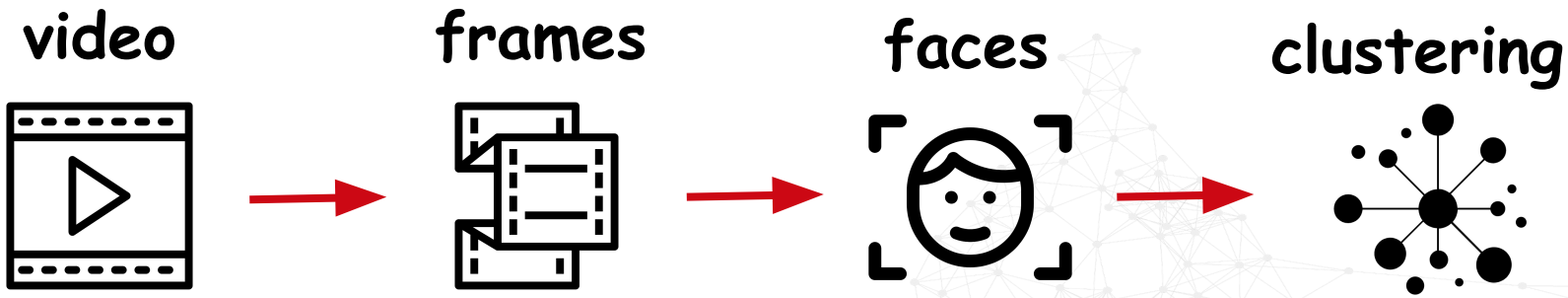
Video search - Boeing OFT news conference



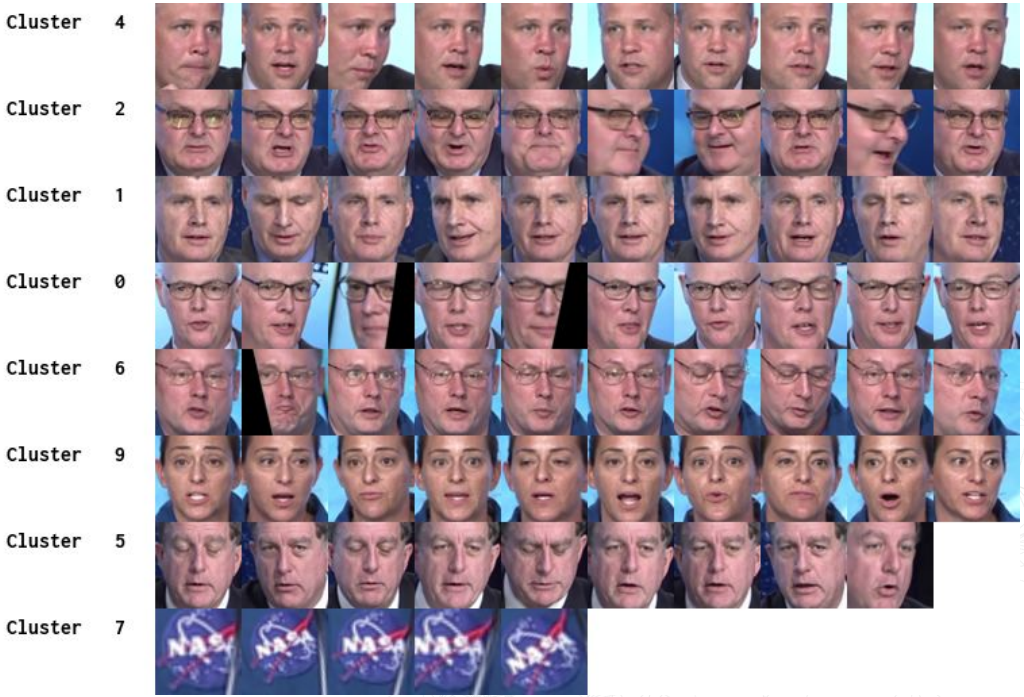
Source images.nasa.gov

Github Repo: github.com/epfl-exts/amld20-video-search - *Lunch Break 12:30*

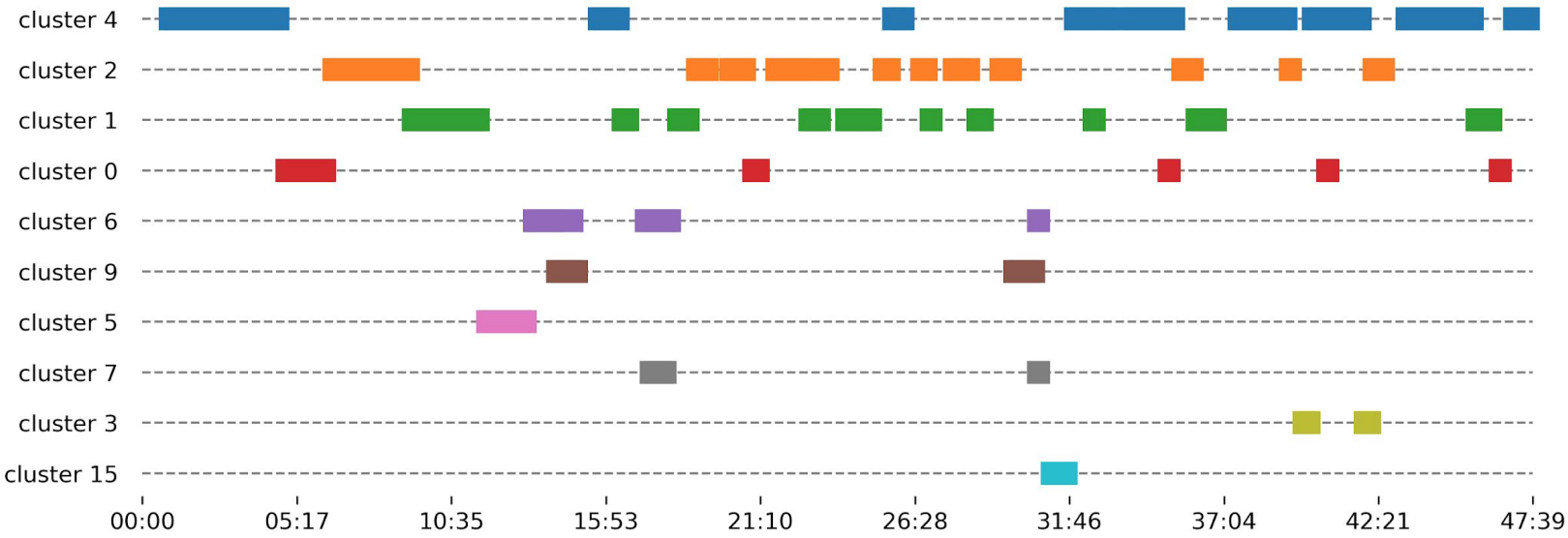
The plan



Goal: identify clusters



Goal: video timeline

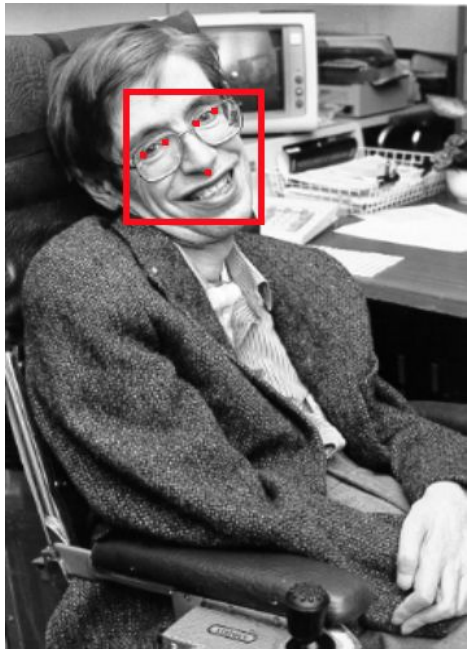


Github Repo: github.com/epfl-exts/amld20-video-search - *Lunch Break 12:30*

GitHub repository

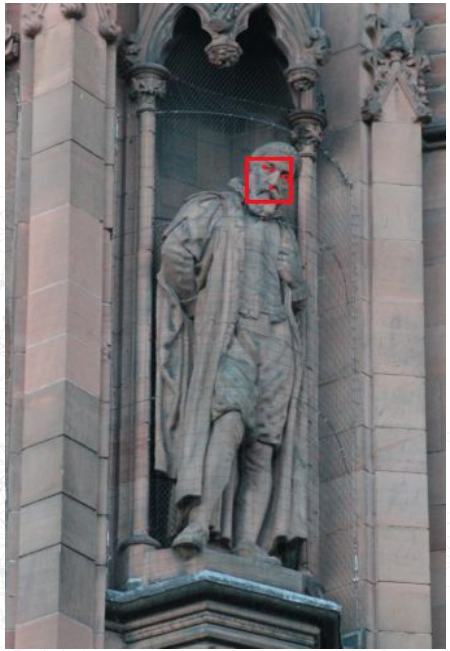
github.com/epfl-exts/amld20-video-search

Face recognition



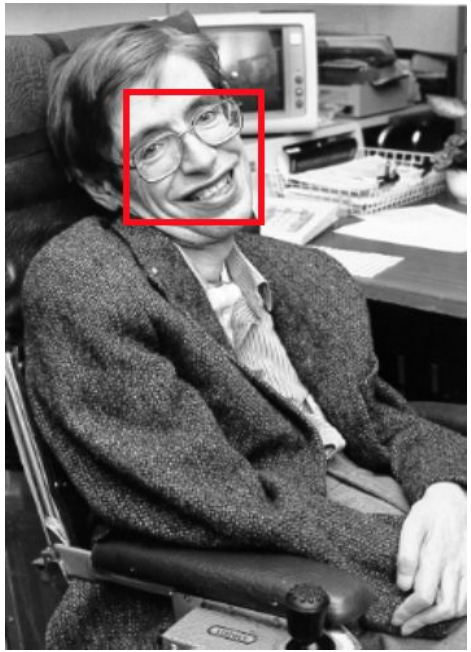
Stephen Hawking

Leonhard Euler

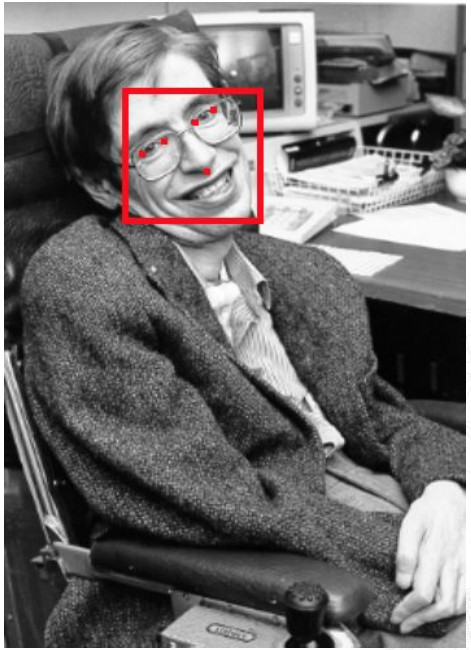


John Napier

Detecting faces



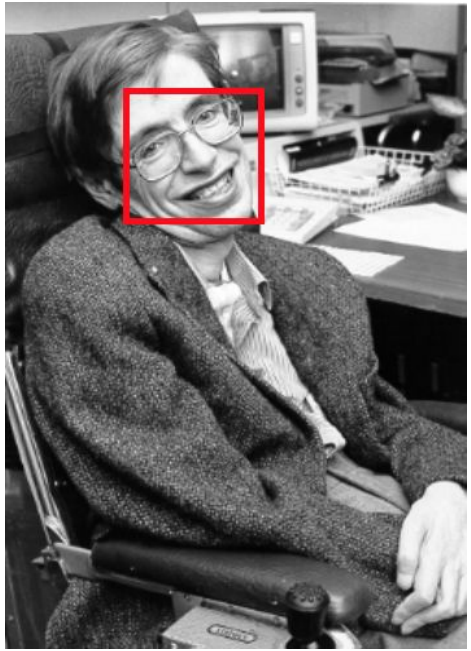
Face detection



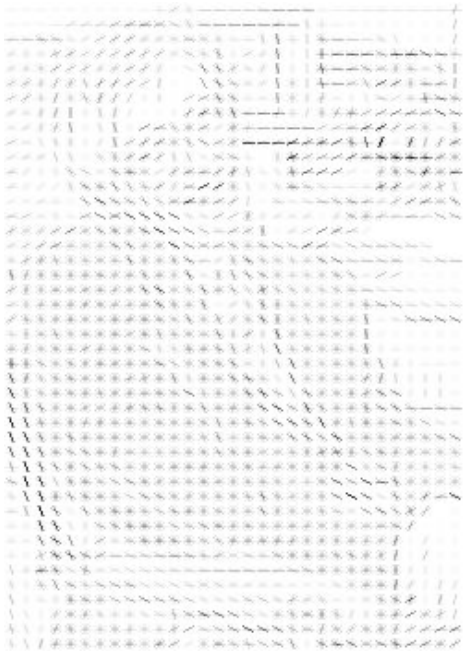
Landmarks

Alignment

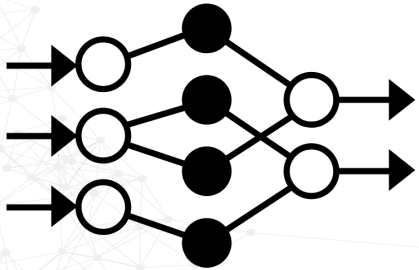
Face detection



Face detection



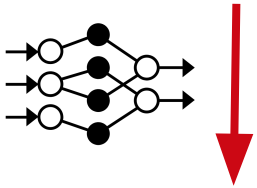
Feature engineering
“hog_detector”



Neural network
“cnn_detector”

Embeddings: changing the feature space

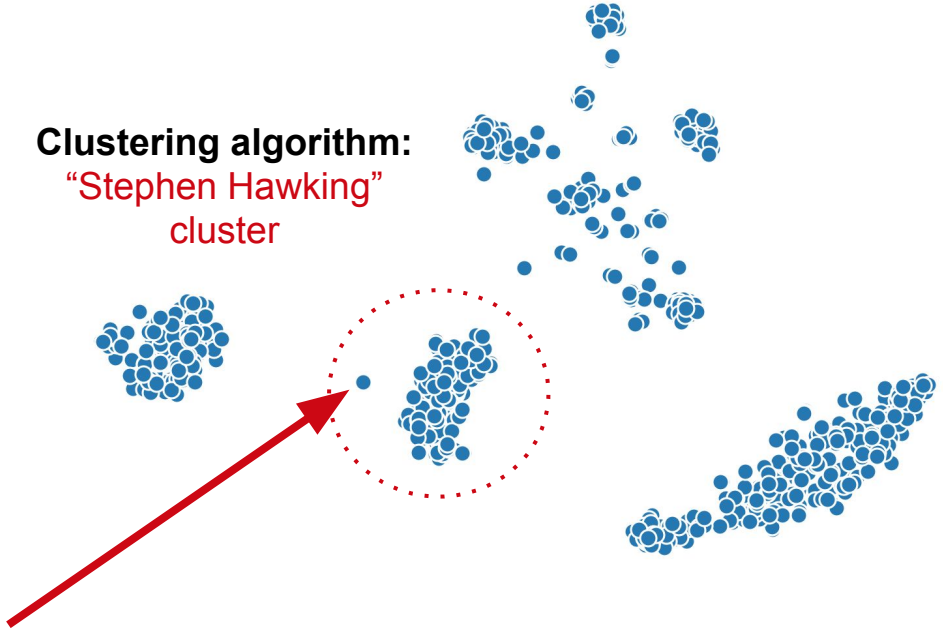
“Pixels” space



“Face embeddings” space

```
face = [0.48, 0.22, 0.17, 0.81,
        ....      0.12, 0.35]
```

Clustering algorithm:
“Stephen Hawking”
cluster



Example: 300 faces - 2d view

Lunch break

12:30 - 13:30