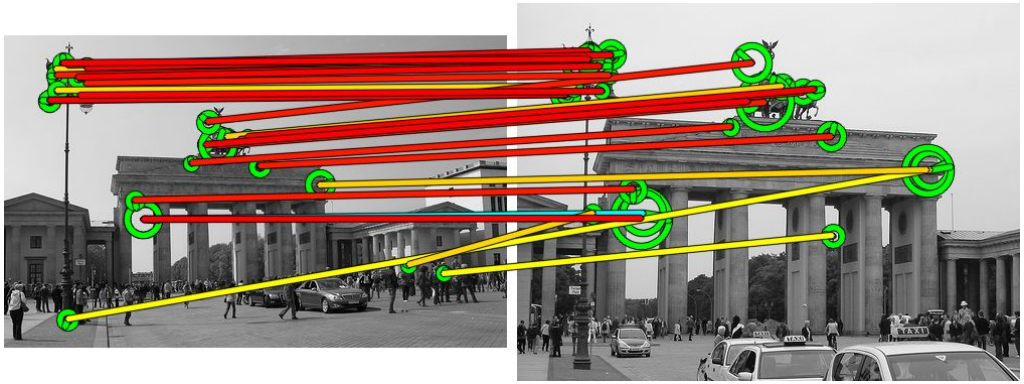
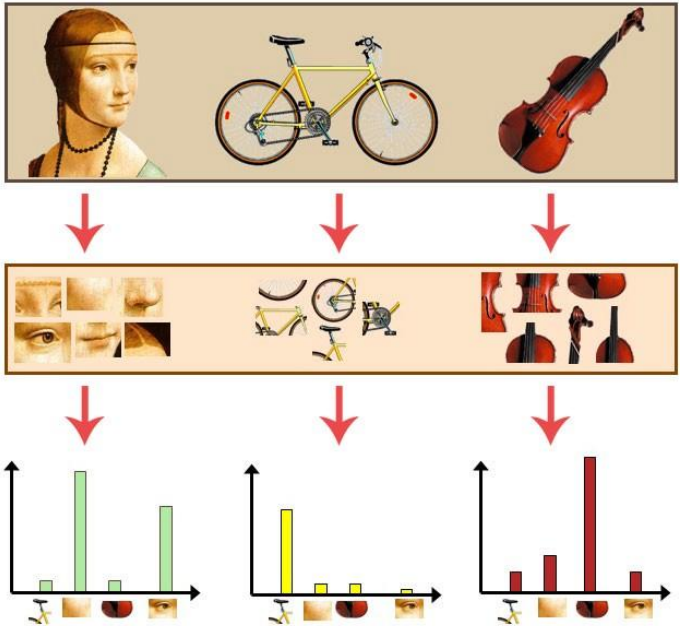
What are the hottest topics in computer vision?

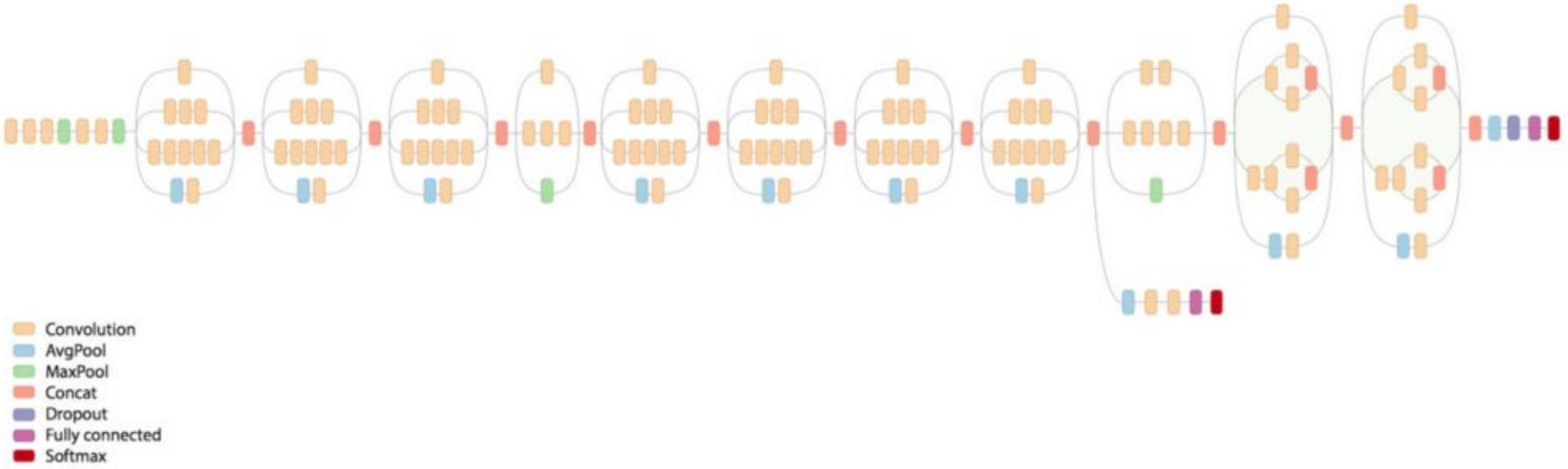
Pablo Suau
Lead Data Scientist @ Partnerize

What it used to be...





What it looks like now...



CVPR 2018 Open Access Repository

CVPR 2018 open access

These CVPR 2018 papers are the Open Access versions, provided by the Computer Vision Foundation.
Except for the watermark, they are identical to the accepted versions; the final published version of the proceedings is available on IEEE Xplore.

Search

Powered by:

Microsoft Azure

Sponsored by:

amazon

facebook

Google

Papers

Embodied Question Answering
Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, Dhruv Batra
[pdf] [supp] [arXiv] [bibtex]

Learning by Asking Questions
Ishan Misra, Ross Girshick, Rob Fergus, Martial Hebert, Abhinav Gupta, Laurens van der Maaten
[pdf] [arXiv] [bibtex]

Finding Tiny Faces in the Wild With Generative Adversarial Network
Yancheng Bai, Yongqiang Zhang, Mingli Ding, Bernard Ghanem
[pdf] [bibtex]

Learning Face Age Progression: A Pyramid Architecture of GANs
Hongyu Yang, Di Huang, Yunhong Wang, Anil K. Jain

CVPR 2018 Open Access Repository

Embodied Question Answering

Abhishek Das[1*], Samyak Datta[1], Georgia Gkioxari[2], Stefan Lee[1], Devi Parikh[2,1], Dhruv Batra[2,1]
[1]Georgia Institute of Technology, [2]Facebook AI Research
[1]{abhshkdz, samyak, steflee}@gatech.edu [2]{gkioxari, parikh, dbatra}@fb.com
embodiedqa.org

Abstract

*We present a new AI task – **Embodied Question Answering** (EmbodiedQA) – where an agent is spawned at a random location in a 3D environment and asked a question ('What color is the car?'). In order to answer, the agent must first intelligently navigate to explore the environment, gather necessary visual information through first-person (egocentric) vision, and then answer the question ('orange').*

EmbodiedQA requires a range of AI skills – language understanding, visual recognition, active perception, goal-driven navigation, commonsense reasoning, long-term memory, and grounding language into actions. In this work, we develop a dataset of questions and answers in House3D environments [1], evaluation metrics, and a hierarchical model trained with imitation and reinforcement learning.

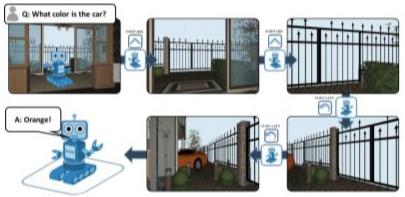


Figure 1: Embodied Question Answering – EmbodiedQA– tasks agents with navigating rich 3D environments in order to answer questions. These agents must jointly learn language understanding, visual reasoning, and goal-driven navigation to succeed.

1. Introduction

with a dataset of questions in virtual environments, evaluation metrics, and a deep reinforcement learning (RL) model.

CVPR 2018 Open Access Repository

Learning by Asking Questions

Ishan Misra[1] * Ross Girshick[2] Rob Fergus[2]
Martial Hebert[1] Abhinav Gupta[1] Laurens van der Maaten[2]
[1]Carnegie Mellon University [2]Facebook AI Research

Abstract

We introduce an interactive learning framework for the development and testing of intelligent visual systems, called learning-by-asking (LBA). We explore LBA in context of the Visual Question Answering (VQA) task. LBA differs from standard VQA training in that most questions are not observed during training time, and the learner must ask questions it wants answers to. Thus, LBA more closely mimics natural learning and has the potential to be more data-efficient than the traditional VQA setting. We present a model that performs LBA on the CLEVR dataset, and show that it automatically discovers an easy-to-hard curriculum when learning interactively from an oracle. Our LBA generated data consistently matches or outperforms the CLEVR train data and is more sample efficient. We also show that our model asks questions that generalize to state-of-the-art VQA models and to novel test time distributions.

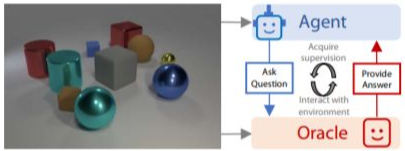


Figure 1: **The Learning-by-Asking (LBA) paradigm.** We present an open-world Visual Question Answering (VQA) setting in which an agent interactively learns by asking questions to an oracle. Unlike standard VQA training, which assumes a fixed dataset of questions, in LBA the agent has the potential to learn more quickly by asking "good" questions, much like a bright student in a class. LBA does not alter the test-time setup of VQA.

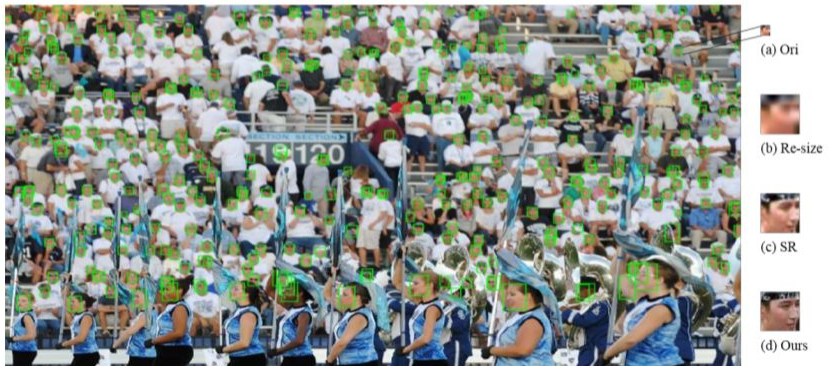
1. Introduction

images and decides *what questions to ask*. Questions asked by the learner are answered by an oracle (human supervi-

CVPR 2018 Open Access Repository

Finding Tiny Faces in the Wild with Generative Adversarial Network

Yancheng Bai[1,3] Yongqiang Zhang[1,2] Mingli Ding[2] Bernard Ghanem[1]
[1] Visual Computing Center, King Abdullah University of Science and Technology (KAUST)
[2] School of Electrical Engineering and Automation, Harbin Institute of Technology (HIT)
[3] Institute of Software, Chinese Academy of Sciences (CAS)
baiyancheng20@gmail.com {zhangyongqiang, dingml}@hit.edu.cn bernard.ghanem@kaust.edu.sa



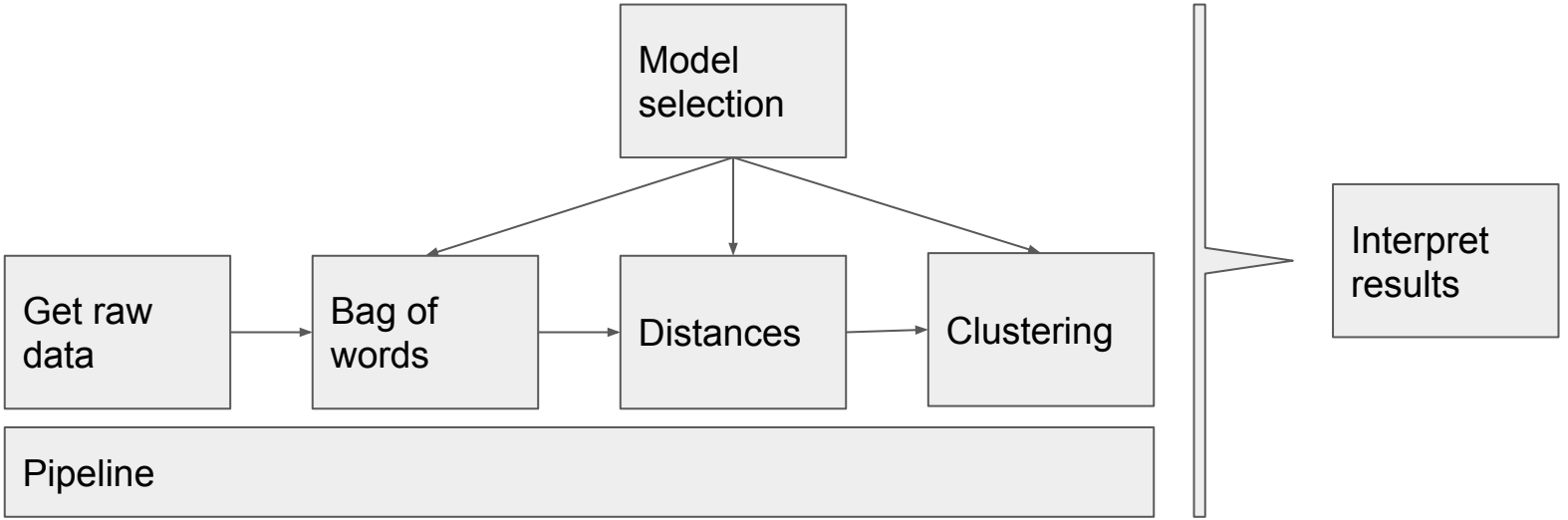
CVPR 2018 Open Access Repository

979 papers!

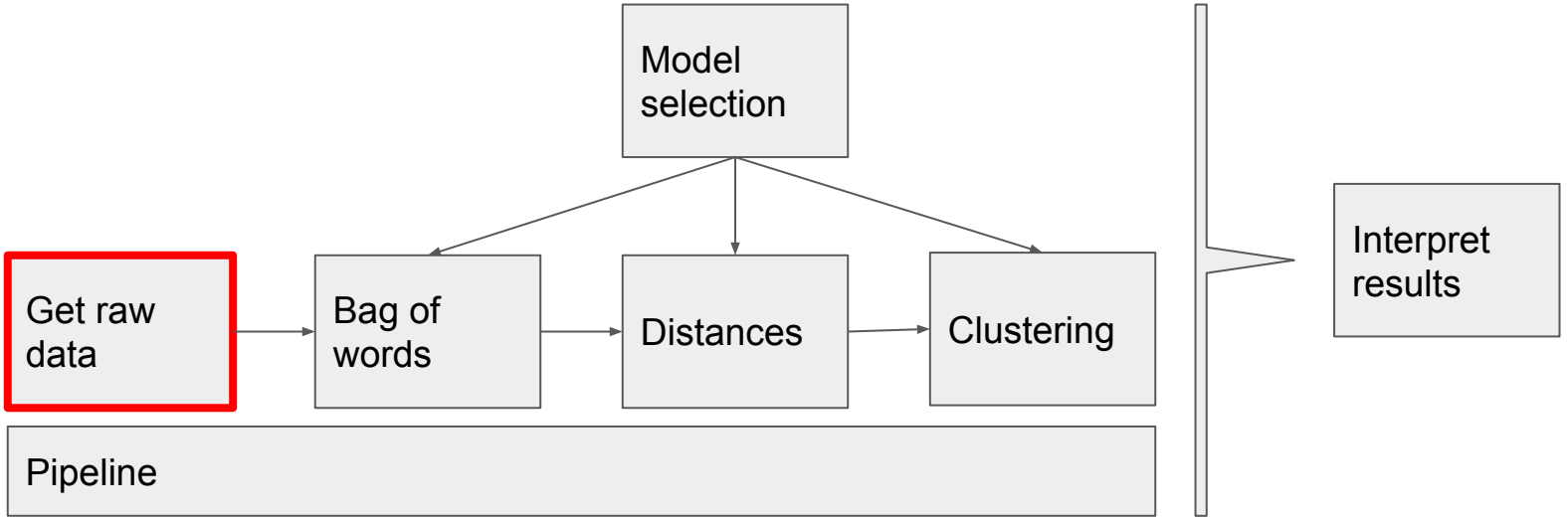
Objectives

- Quick rough idea
- Overcome bad habits (BeautifulSoup, scikit.learn)

The process


Model selection
Get raw data
Bag of words
Distances
Clustering
Interpret results
Pipeline

The process


Model selection
Get raw data
Bag of words
Distances
Clustering
Interpret results
Pipeline

Getting the raw data

In [ ]:
```
# Getting the page
response = requests.get("http://openaccess.thecvf.com/CVPR2018.py")
page_html = response.text
soup = BeautifulSoup(page_html, 'html.parser')
```

In [ ]:
```
# Getting the name of the pdf files
pattern = re.compile(r'pdf')
urls = ['http://openaccess.thecvf.com/' + element.get('href') for element in soup.find_all('a', text=patte
rn)]
print('Number of papers: ' + str(len(urls)))
print('First 5 urls: ')
for u in urls[0:5]:
    print('    ' + u)
```

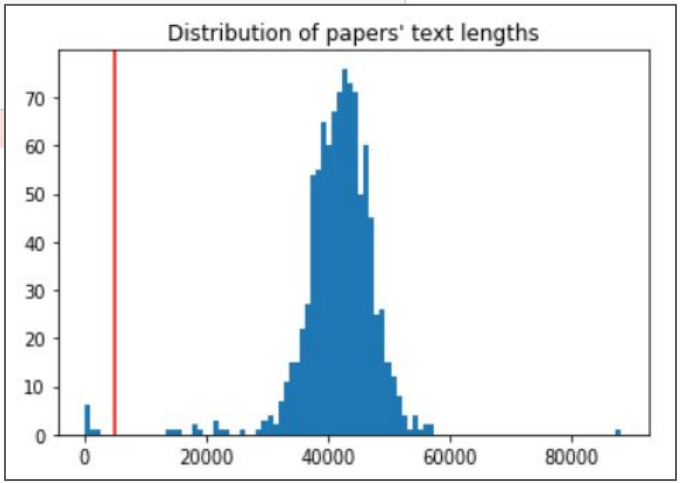
In [ ]:
```
if not os.path.exists('papers/'):
    os.makedirs('papers/')

for i in tqdm(range(len(urls))):
    url = urls[i]
    file = 'papers/' + os.path.basename(url)
    response = requests.get(url, stream=True)
    with open(file, 'wb') as handle:
        for data in response.iter_content():
            handle.write(data)
```

Getting the raw data

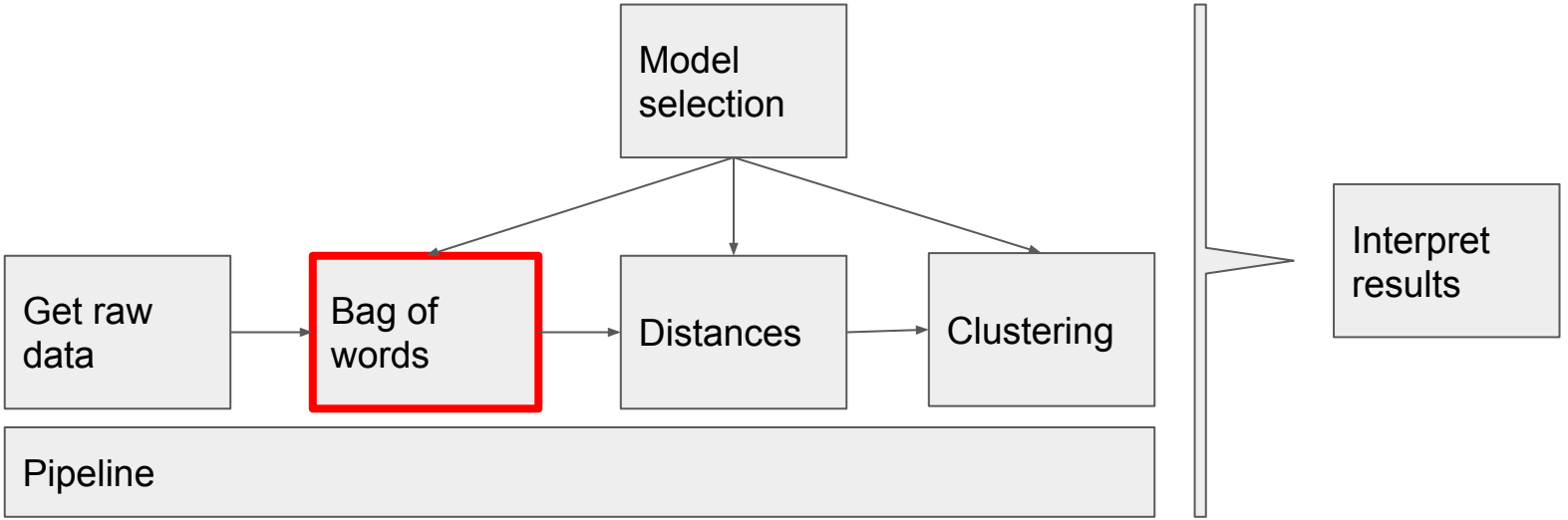
In [4]:
```
# Extract the raw text from the papers
papers = sorted(glob.glob('papers/*.pdf')) # Sorting alphabetically makes
                                           # debugging easier

for i in tqdm(range(len(papers))):
    paper = papers[i]
    output_file = os.path.join('data/', os.path.basename(paper)).replace('.pdf', '.txt')
    if not os.path.exists(output_file):
        text = convert_pdf_to_txt(paper)
        try:
            with open(output_file, 'w') as f:
                f.write(text)
        except:
            os.remove(output_file)
```

```
100%|██████████| 975/975 [00:00<00:00, 100774.92it/s]
```

pdfminer2



The process


Model selection
Get raw data
Bag of words
Distances
Clustering
Interpret results
Pipeline

Bag of words model

In [4]:
```
count_vect = TfidfVectorizer(input='filename', max_df = 0.7, min_df = 0.3)
X_freq = count_vect.fit_transform(papers)
X_freq.shape
```

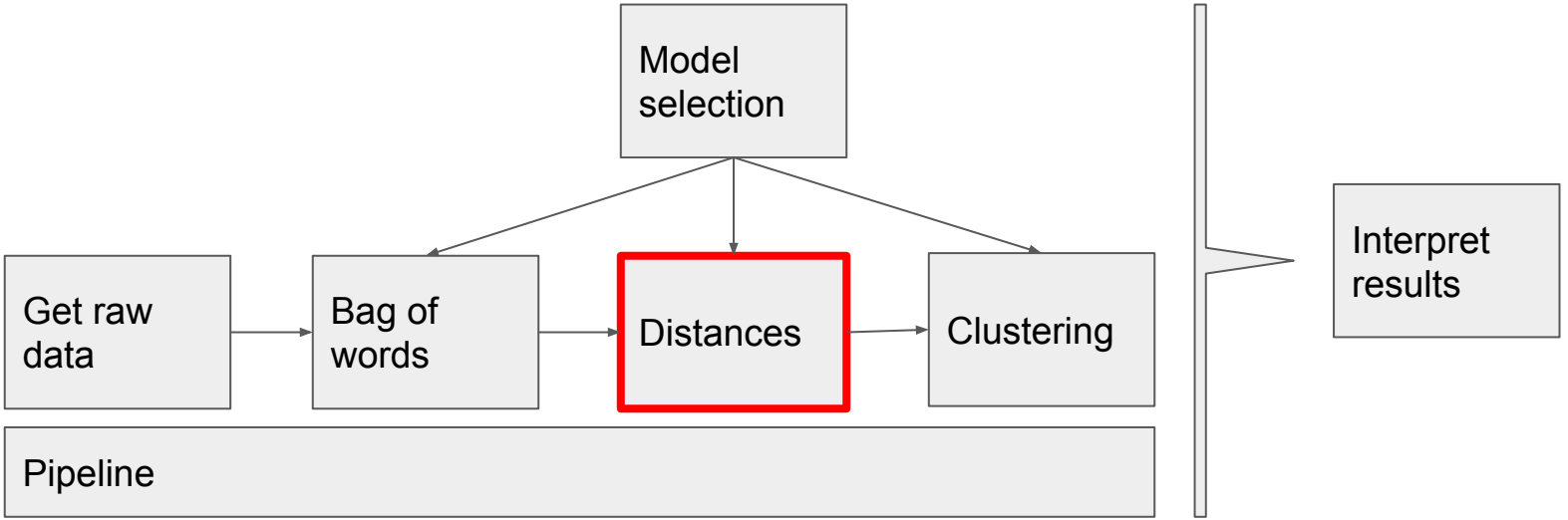
Out[4]: (966, 867)

In [6]:
```
X_freq[0,1]
```

Out[6]: 0.039063045941197076

```
'metric': 518,
'metrics': 519,
'might': 520,
'min': 521,
'minimize': 522,
'minimizing': 523,
'modeling': 524,
'module': 525,
'moreover': 526,
'motion': 527,
'national': 528,
'natural': 529,
'nature': 530,
'necessary': 531,
'need': 532,
'needs': 533,
'negative': 534,
'net': 535,
'nets': 536,
'next': 537,
'nips': 538,
'noise': 539,
'normalization': 540,
'normalized': 541,
'novel': 542,
```

The process


Model selection
Get raw data
Bag of words
Distances
Clustering
Pipeline
Interpret results

Computing similarities

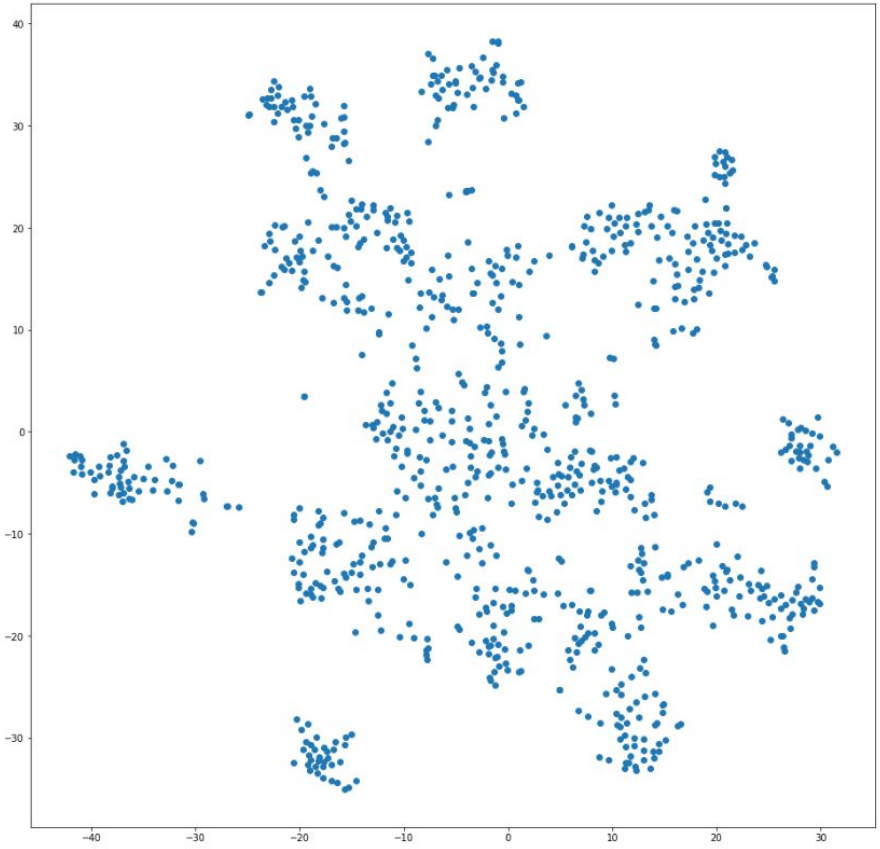
LSA

In [7]:
```
X_lsa = TruncatedSVD(n_components=15, random_state=0).fit_transform(X_freq)
```

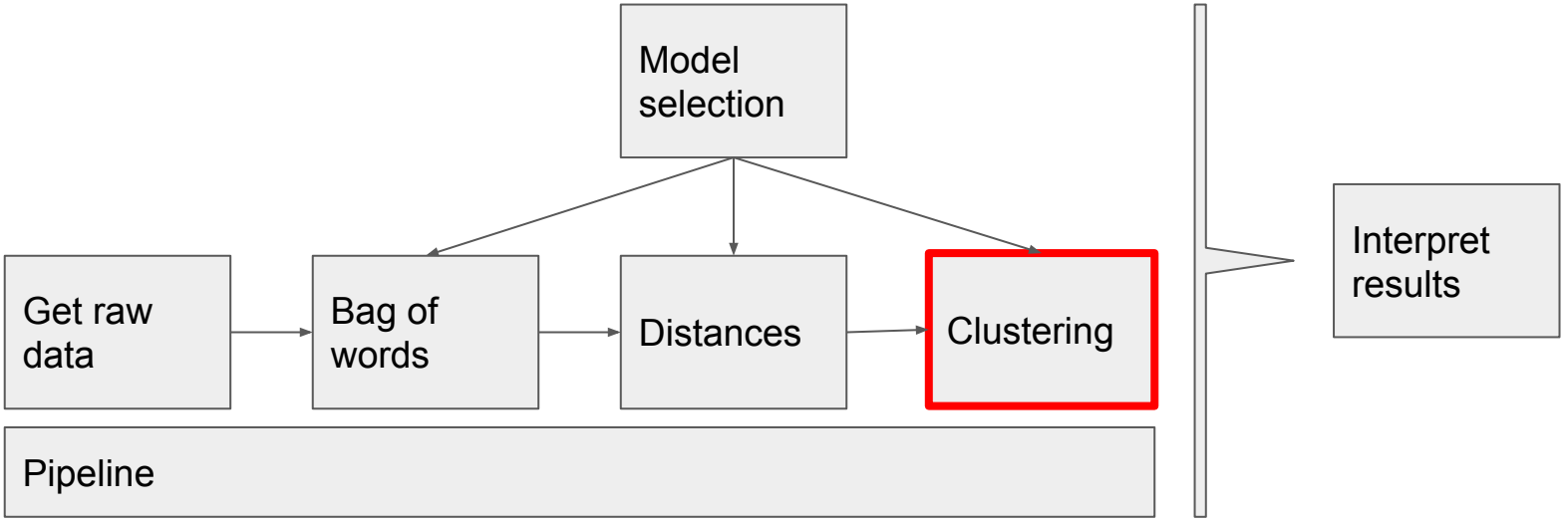
Paper similarity

In [8]:
```
X_embedded = TSNE(n_components=2).fit_transform(X_lsa)
```

Computing similarities



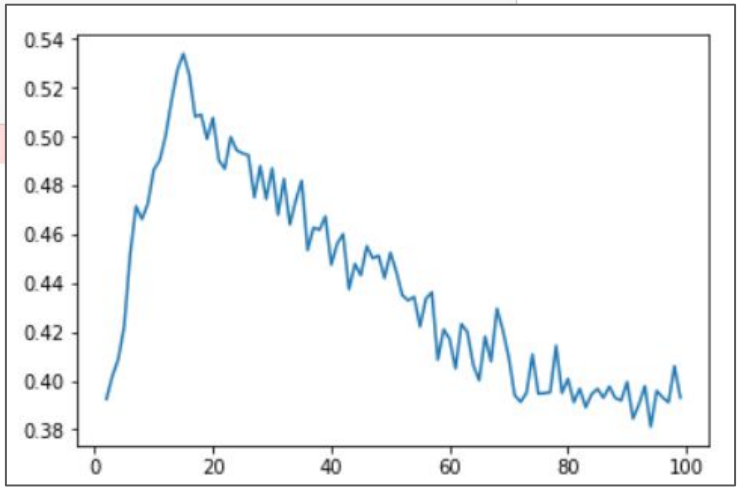
The process


Model selection
Get raw data
Bag of words
Distances
Clustering
Interpret results
Pipeline

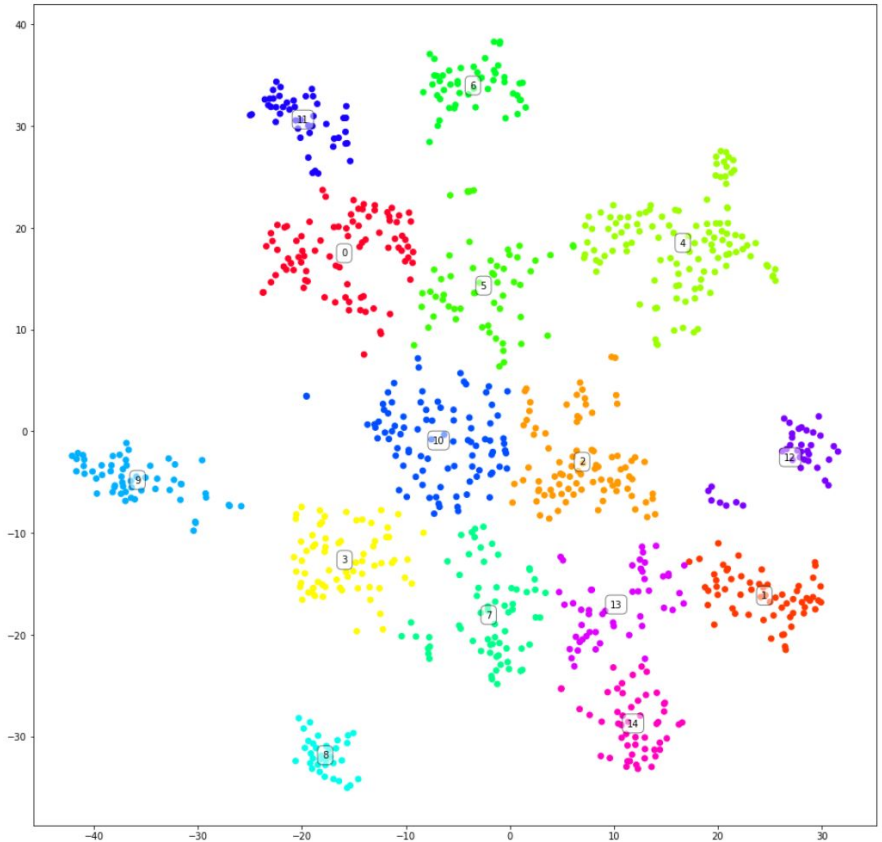
Clustering

In [10]:
```
# Model selection
RANGE_K = range(2, 100)
best_kmeans = None
max_shil = -1
shils = []
for k in tqdm(RANGE_K):
    kmeans = KMeans(n_clusters=k, random_state=0).fit(X_embedded)
    shil = silhouette_score(X_embedded, kmeans.labels_)
    shils.append(shil)
    if shil > max_shil:
        max_shil = shil
        best_kmeans = kmeans
kmeans = best_kmeans
```

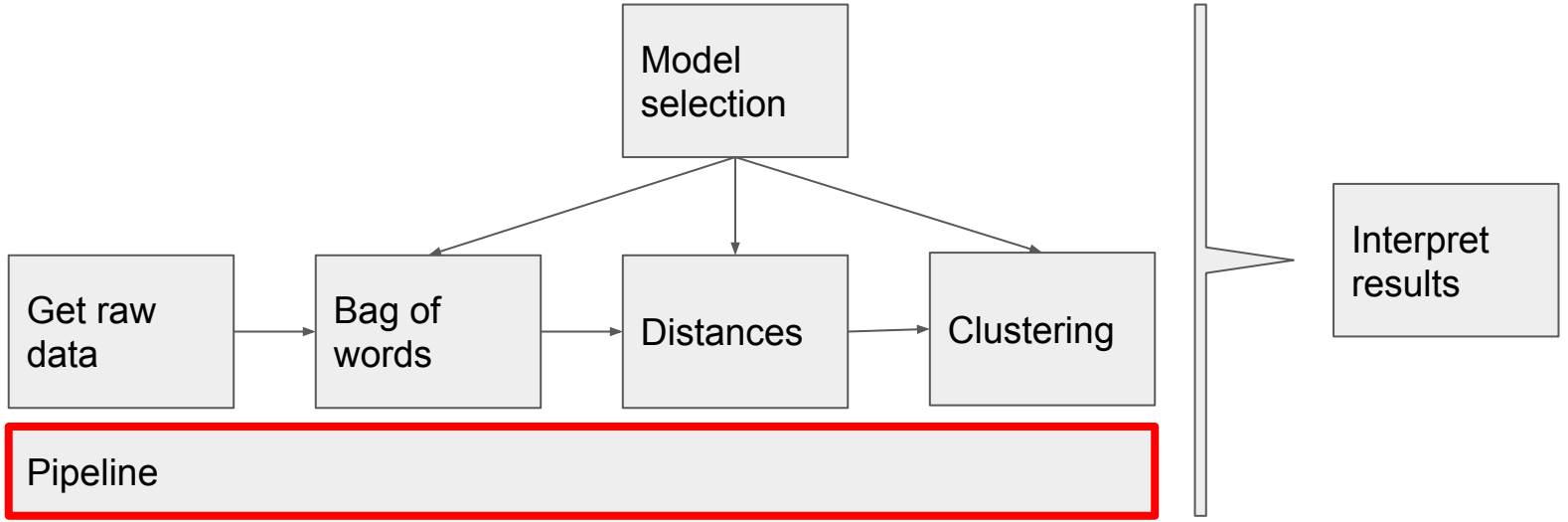
```
100%|██████████| 98/98 [00:24<00:00,  4.03it/s]
```



Clustering


40
30
20
10
0
−10
−20
−30
−40
−30
−20
−10
0
10
20
30
11
6
0
4
5
10
2
12
9
3
13
7
1
14
8

The process

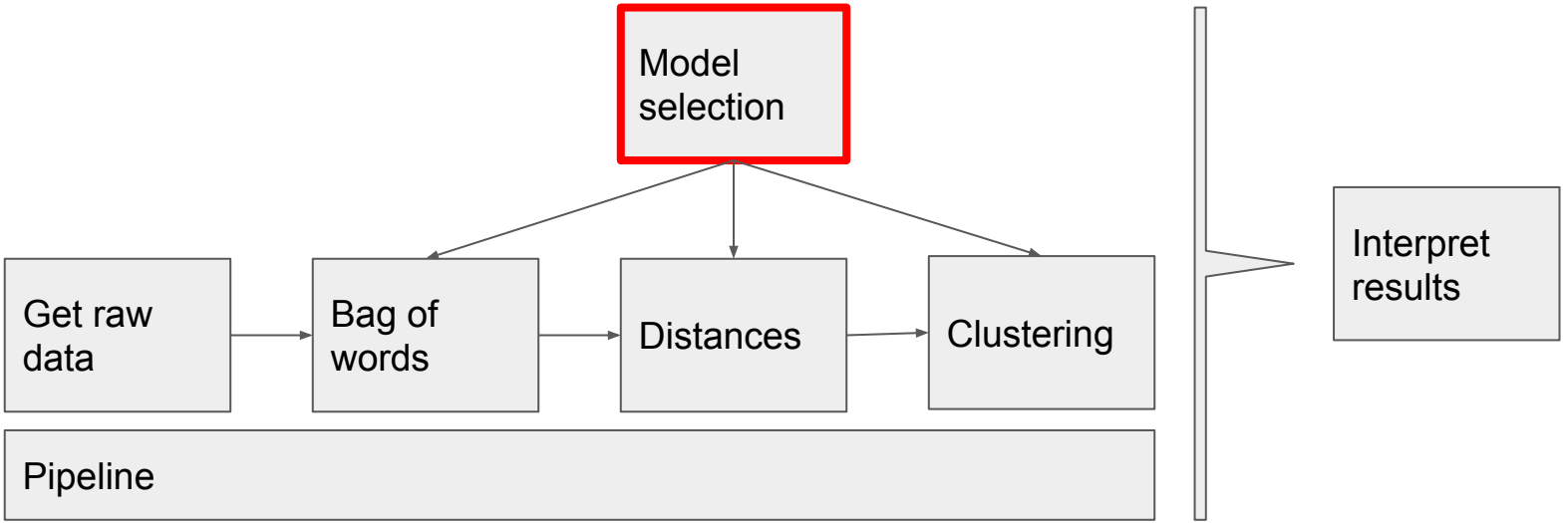

Model selection
Get raw data
Bag of words
Distances
Clustering
Interpret results
Pipeline

Putting all together

In [11]:
```
def pipeline_kmeans(input_data):
            df = input_data[0]
            max_df = input_data[1]
            min_df = input_data[2]
            n_components = input_data[3]
            k = input_data[4]

            X_freq = build_bag_of_words(df, max_df = max_df, min_df = min_df)
            X_lsa = apply_lsa(X_freq, n_components = n_components)
            X_embedded = apply_dimensionality_reduction(X_lsa)
            kmeans, shil_kmeans = kmeans_clustering(X_embedded, k = k)
            return [shil_kmeans, kmeans, max_df, min_df, n_components, k, X_embedded]
```

Hyperparameters

The process


Model selection
Get raw data
Bag of words
Distances
Clustering
Interpret results
Pipeline

Model selection

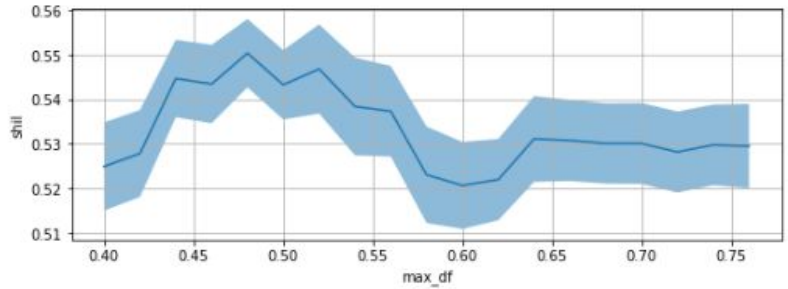
In [13]:
```
max_df_values = np.arange(0.7, 1, 0.1)
min_df_values = np.arange(0, 0.3, 0.1)
n_components_values = range(5, 50, 10)
k_values = range(2, 50, 5)
```

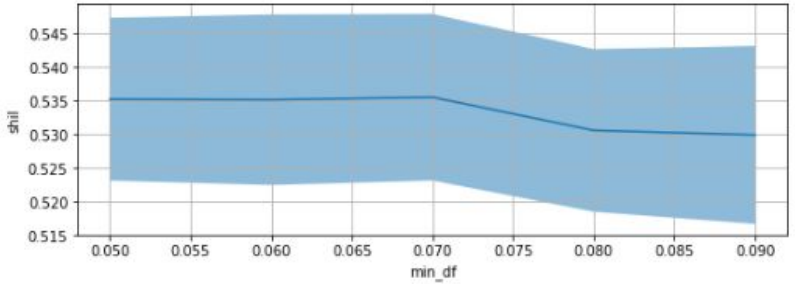
In [14]:
```
grid_search_kmeans(max_df_values,
                   min_df_values,
                   n_components_values,
                   k_values,
                   'checkpoints/coarse.csv')
```

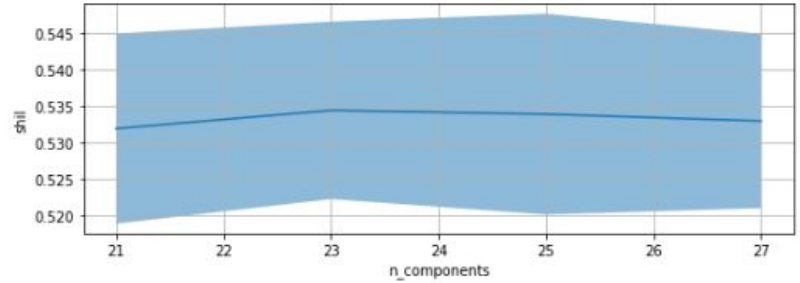
Model selection

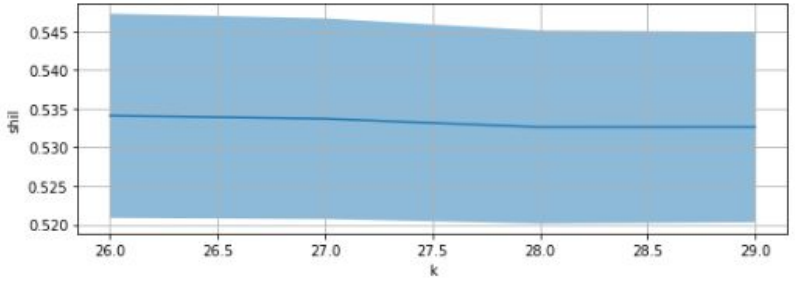
Out[17]:

	index	shil	max_df	min_df	n_components	k
0	505	0.577503	0.52	0.06	25.0	26.0
1	506	0.574212	0.52	0.06	25.0	27.0
2	281	0.566184	0.46	0.07	25.0	26.0
3	201	0.565891	0.44	0.07	25.0	26.0
4	493	0.565487	0.52	0.05	27.0	26.0

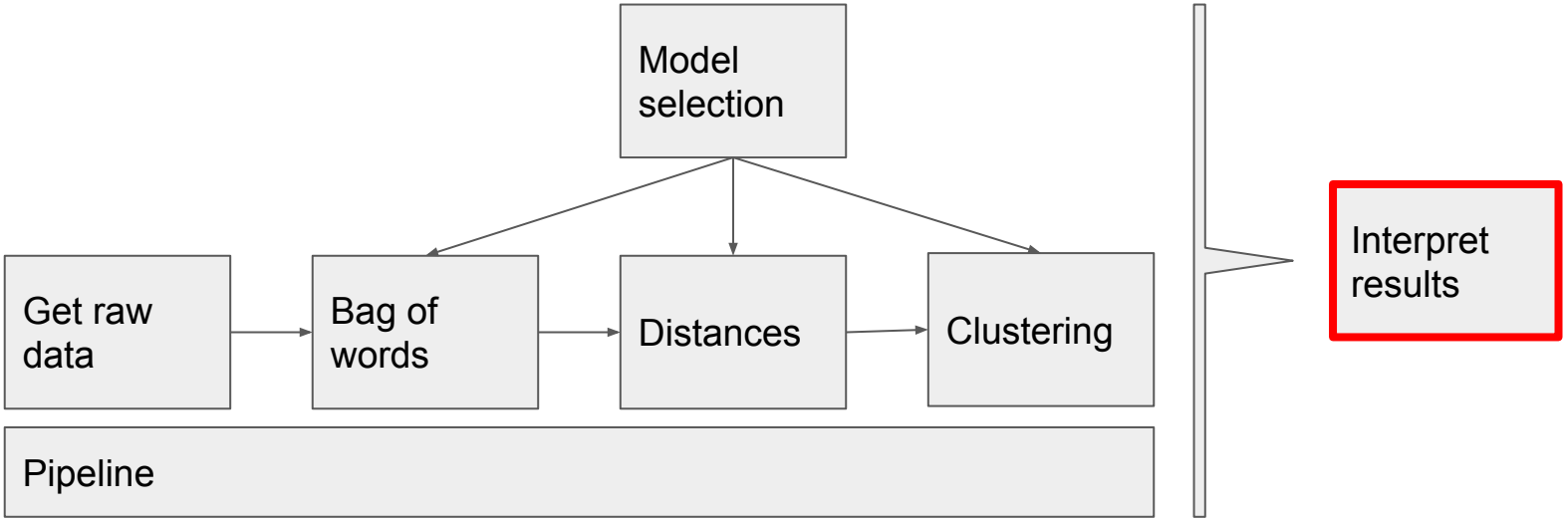




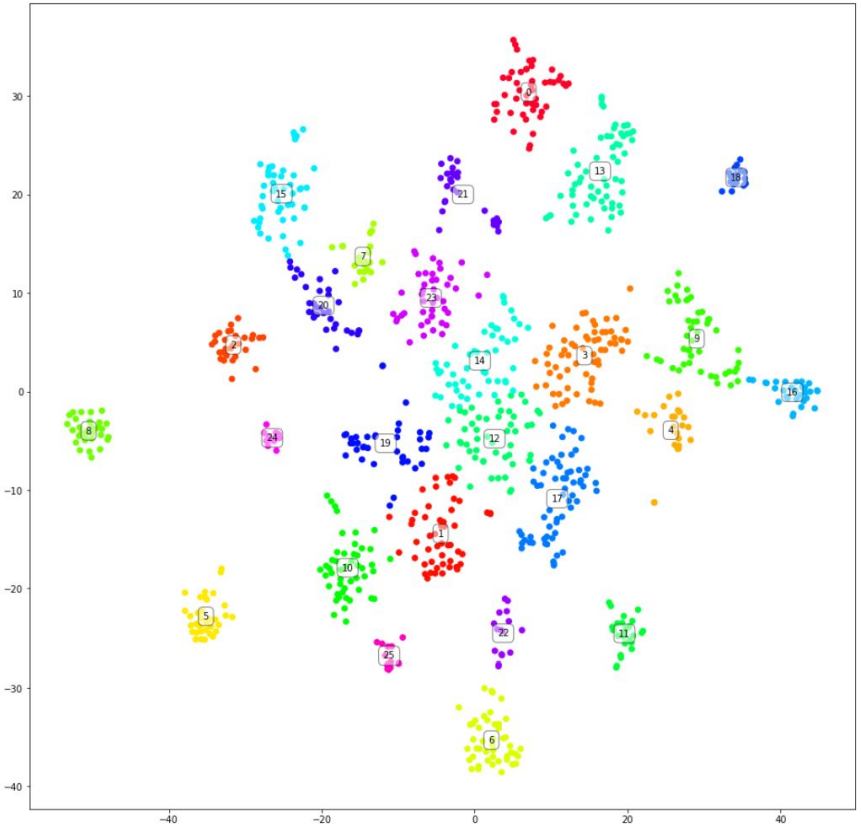




The process

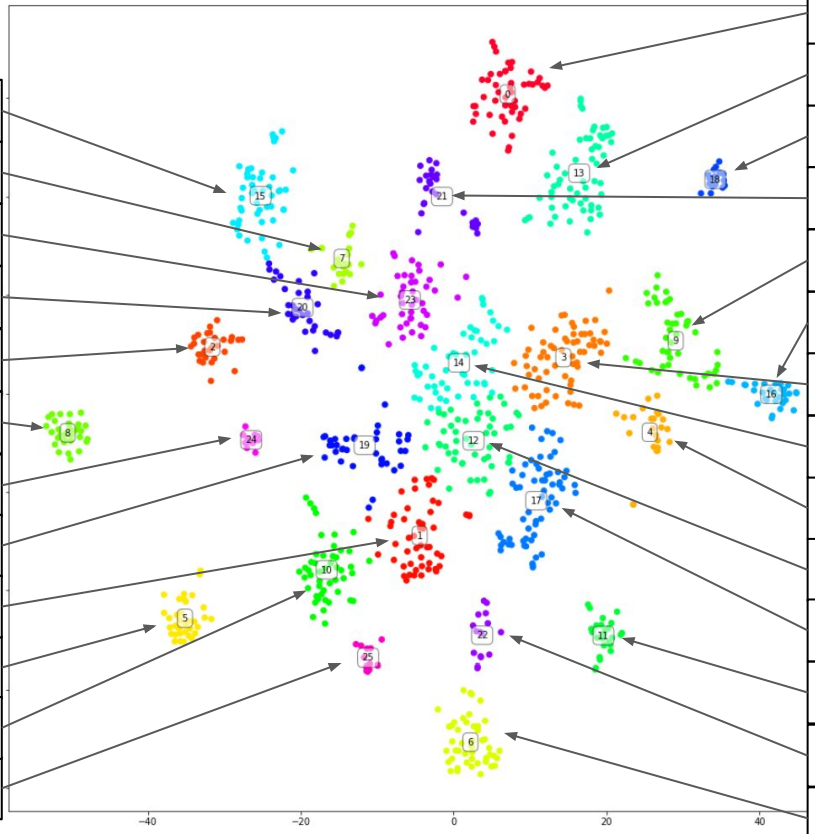

Model selection
Get raw data
Bag of words
Distances
Clustering
Pipeline
Interpret results

Clustering results



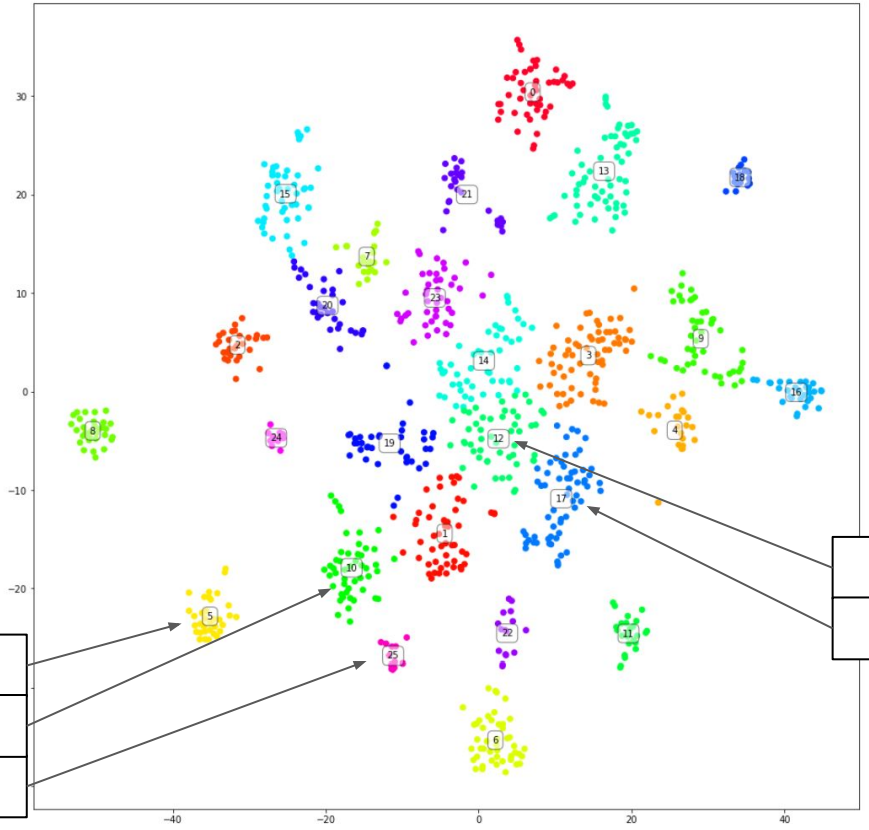
And the topics?

- Pose estimation
- 3D objects
- Point clouds matching
- 3D reconstruction
- SLAM and 3D reconstruction
- Re-identification
- Stereo
- Lighting models
- Image processing
- Transfer learning and adversarial nets
- Generative Adversarial Networks (GAN)
- Generative Adversarial Networks (GAN)



- Video object detection, video processing
- Temporal representation/prediction
- Visual saliency and gaze detection
- Tracking
- Image/video captioning, object detection
- Visual questions answering
- Object localisation/detection
- Image/points registration, visual features
- Text detection/recognition
- Deep learning
- Neural networks
- Object recognition from few examples
- Face detection/classification
- Face detection/recognition

The real impact of deep learning



Deep learning

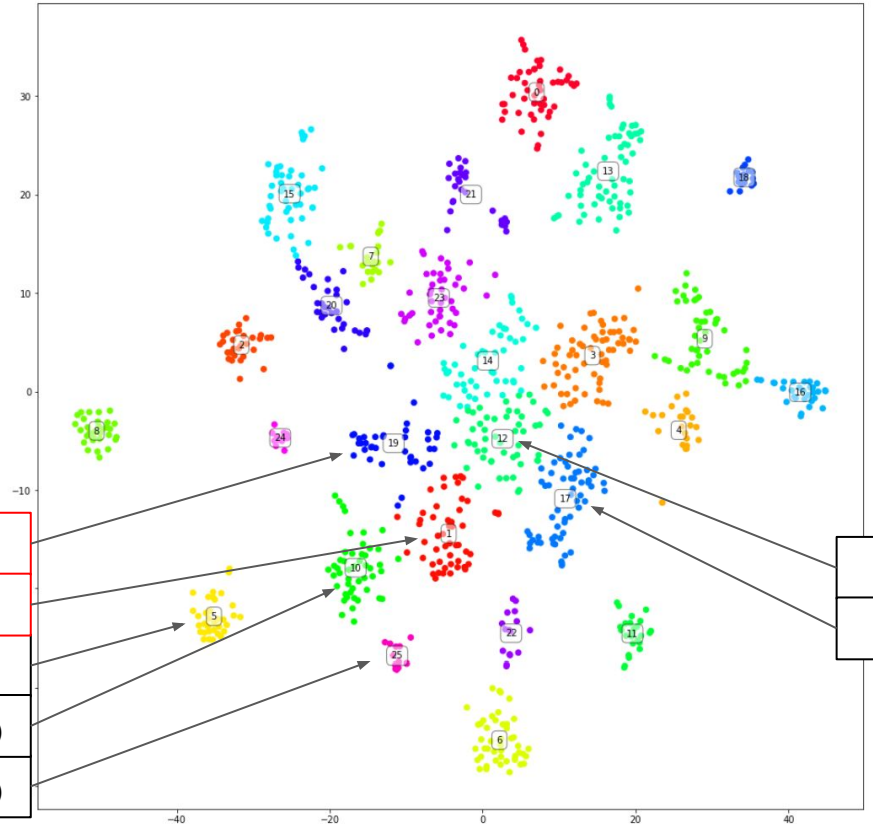
Neural networks

Transfer learning and adversarial nets

Generative Adversarial Networks (GAN)

Generative Adversarial Networks (GAN)

What's in between?



Lighting models

Image processing

Transfer learning and adversarial nets

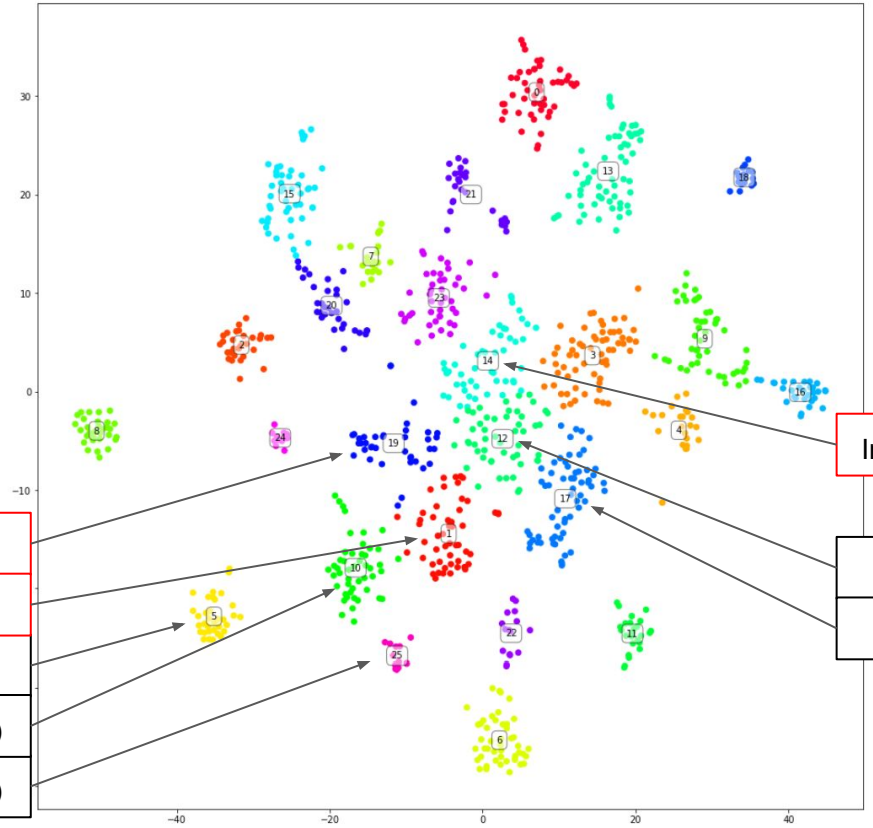
Generative Adversarial Networks (GAN)

Generative Adversarial Networks (GAN)

Deep learning

Neural networks

What about visual features?



Lighting models

Image processing

Transfer learning and adversarial nets

Generative Adversarial Networks (GAN)

Generative Adversarial Networks (GAN)

Image/points registration, visual features

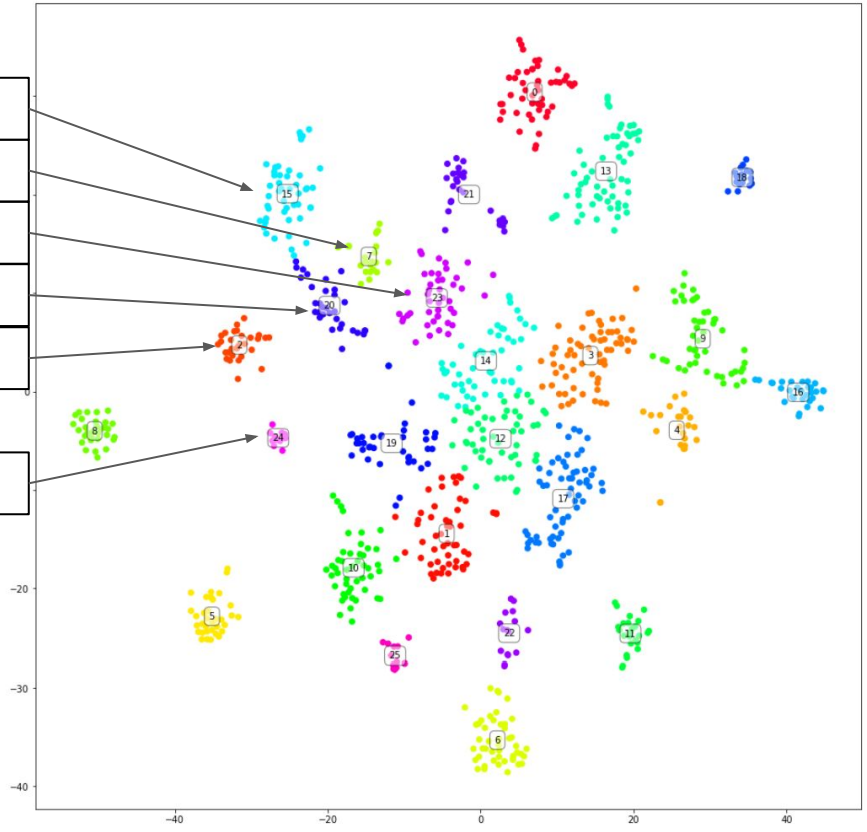
Deep learning

Neural networks

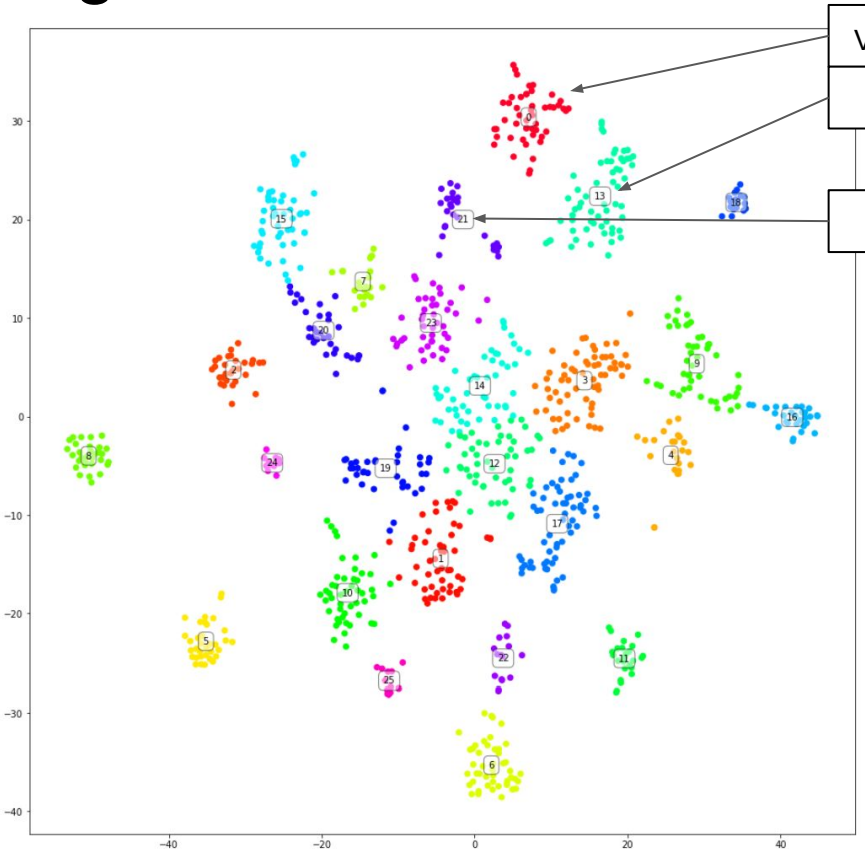
3D imaging

- Pose estimation
- 3D objects
- Point clouds matching
- 3D reconstruction
- SLAM and 3D reconstruction

- Stereo



Video processing

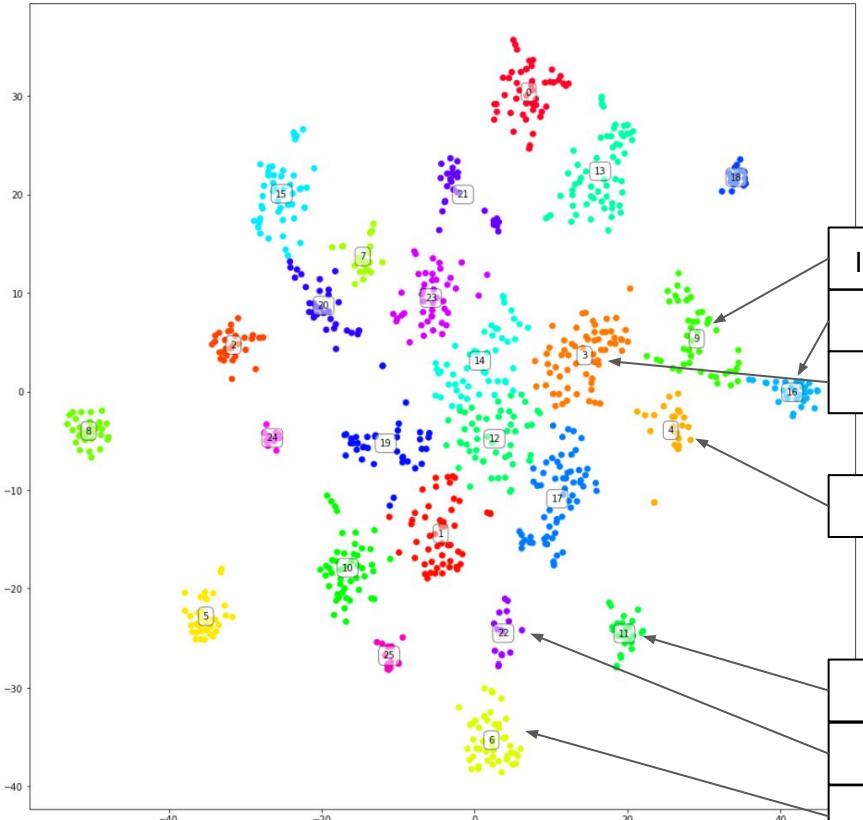


Video object detection, video processing

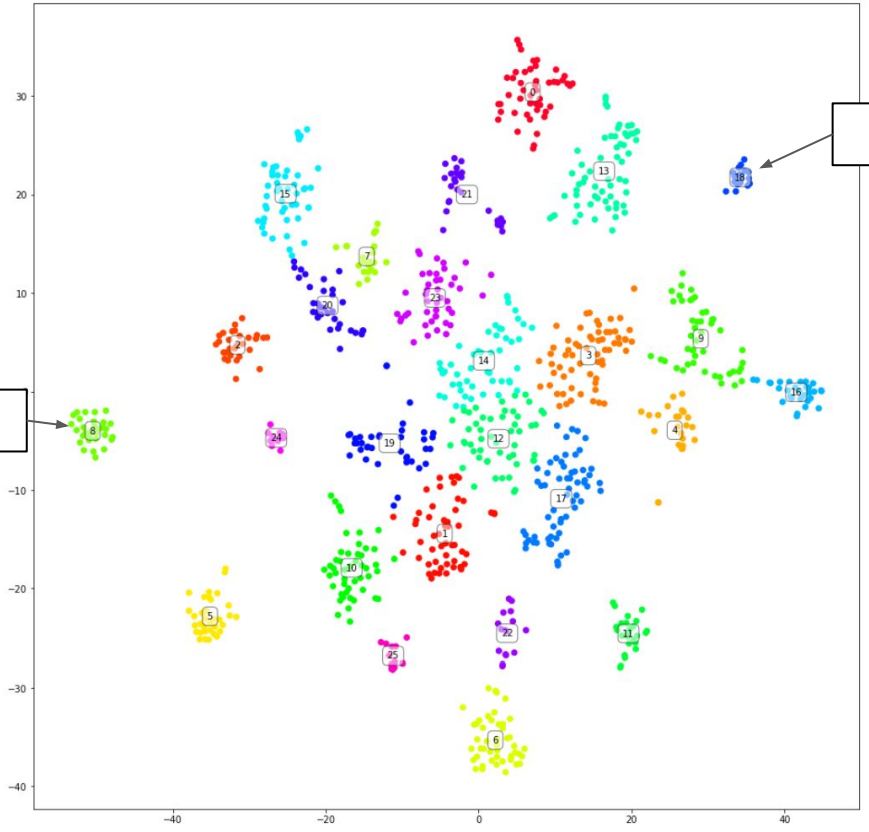
Temporal representation/prediction

Tracking

Object detection/recognition



Other


Visual saliency and gaze detection
Re-identification
30
20
10
−10
−20
−30
−40
−40
−20
0
20
40
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

What are the hottest topics in computer vision?

Pablo Suau
Lead Data Scientist @ Partnerize


pablosuau@gmail.com


https://github.com/pablosuau