Using object detection for hotel pictures classification with CNTK

Karol Żak

Software Engineer

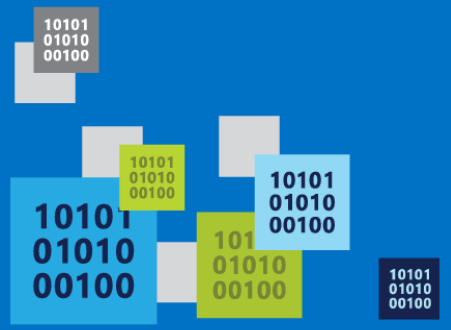
Commercial Software Engineering

Microsoft | Central Eastern Europe

karol.zak@microsoft.com

@KarolZak13

karolzak



Microsoft

Hotailor | TAILORED HOTEL BOOKINGS

- Multi-awarded travel tech startup company from Poland (on BizSpark+)
- Tailored hotel offers mostly for B2B (www.hotailors.com) but also for B2C (www.hotailor.com)
- SaaS solution for traditional travel retailers to maximize profit on hotel bookings
- 90% of travel agencies are still doing bookings manually via email requests or phones!
- Using Hotailors, travel agents are able to handle a process from accommodation request to confirmation voucher within 5 minutes
- Thanks to Hotailors, travel agents can finalize up to 10 times more bookings than manually allowing them to compete with the biggest online booking offers providers and still keep the personal relationship with the client

Microsoft

Challenge

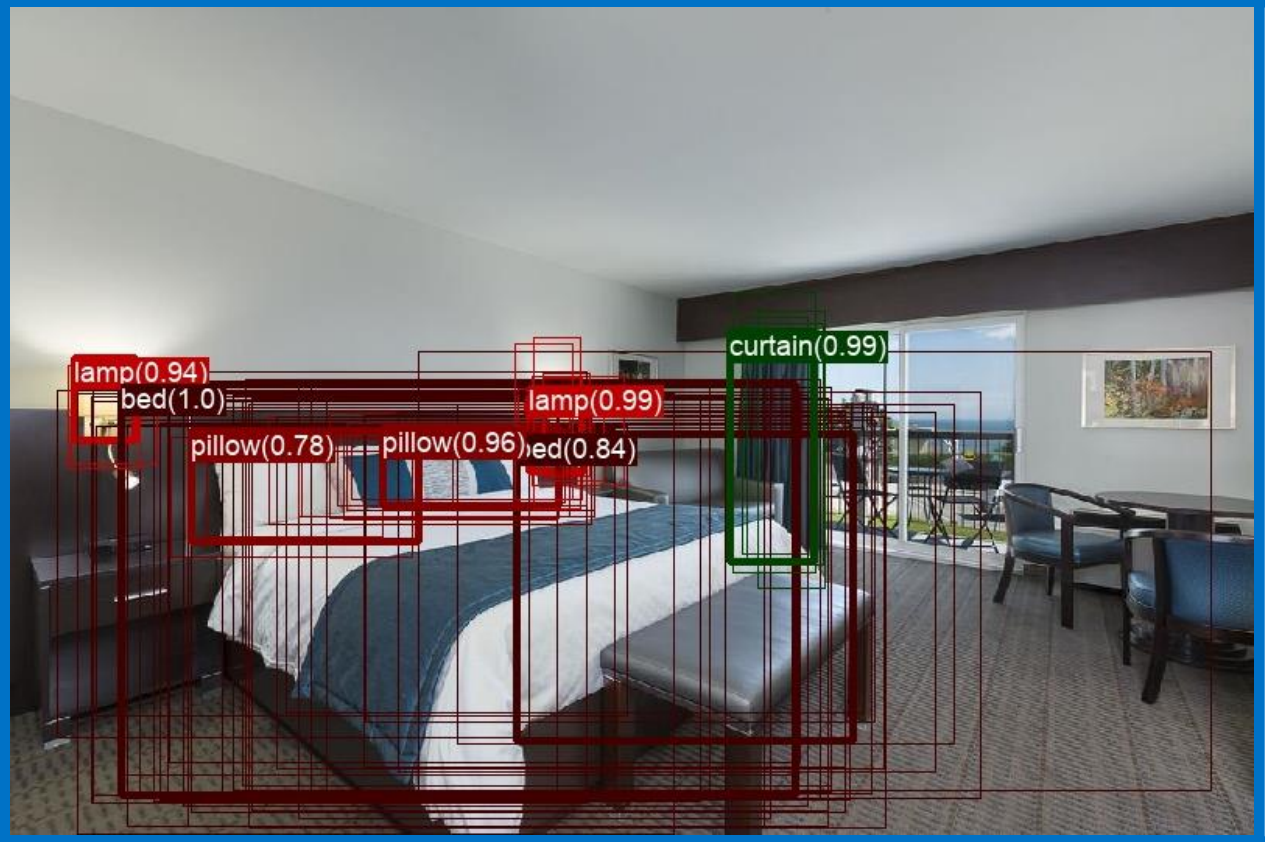
- People buy with their eyes so hotel offers should be simple and show as much as possible with least amount of pictures
- 50+ million pictures (from multiple sources) for almost 1 million different hotels in 198 countries
- Classify pictures for each hotel and prepare certain types of pictures in specific order, i.e. Hotel front -> Reception -> Bedroom -> Bathroom… etc.

Microsoft

Solution

- Use deep learning to detect objects related to certain types of pictures, i.e.:
- Based on detected objects decide what type of picture/room it is

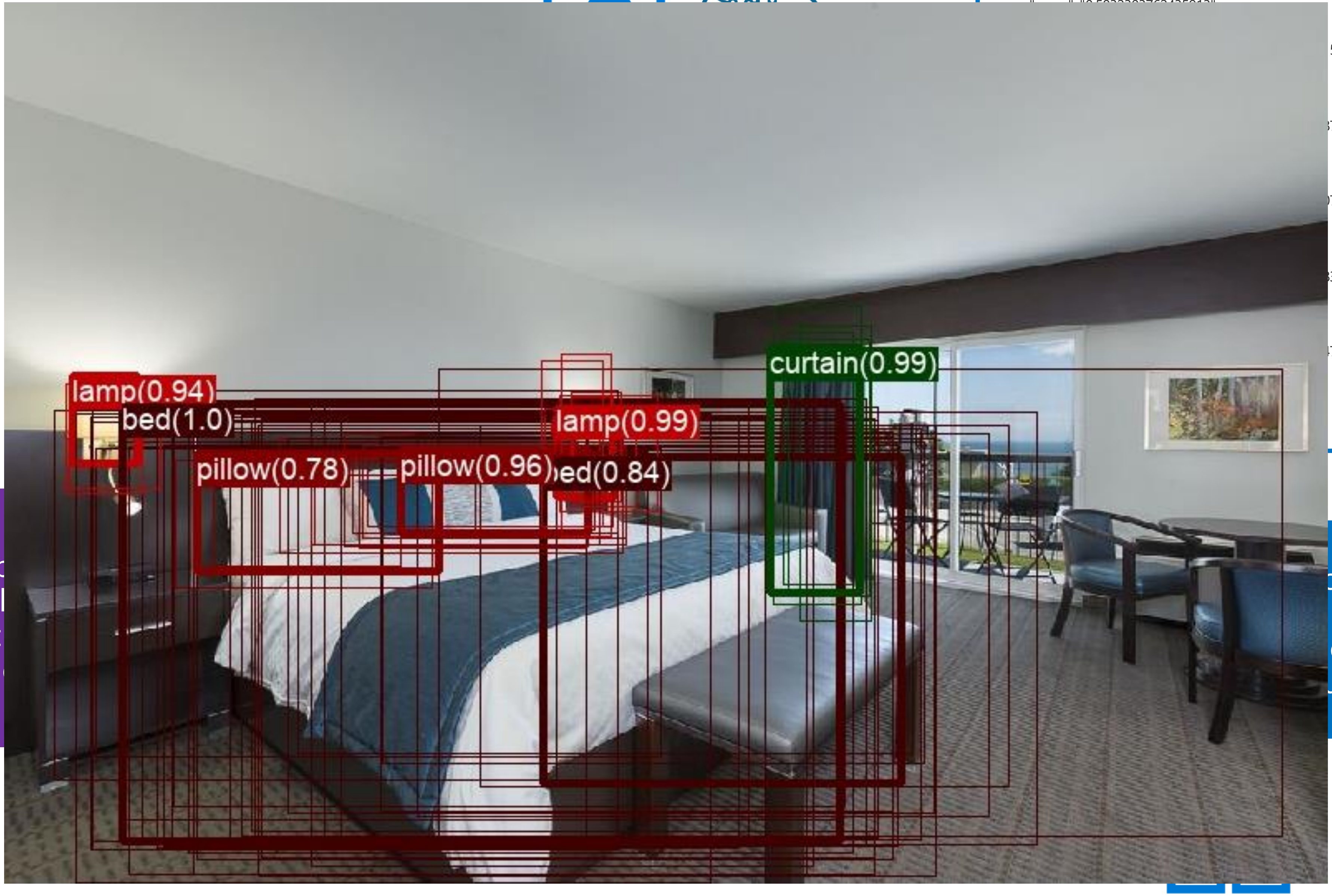
bedroom	bathroom
pillow	tap
bed	sink
curtain	towel
lamp	toilet




```
{
  "tags": [
    {
      "bbox": "[ 168.90582275  493.13183594  299.44207764  609.17419434]",
      "label": "pillow",
      "score": "0.5832303762435913"
    },
    {
      "bbox": "[ 205.11161804  470.61270142  515.55615234  707.32818604]",
      "label": "bed",
      "score": "0.694624662399292"
    },
    {
      "bbox": "[ 417.20318604  477.86273193  451.921875    502.35699463]",
      "label": "lamp",
      "score": "0.7596455812454224"
    },
    {
      "bbox": "[ 592.43994141  358.20336914  649.5020752   648.09082031]",
      "label": "curtain",
      "score": "0.9054949879646301"
    },
    {
      "bbox": "[ 244.92529297  488.33349609  747.13983154  832.        ]",
      "label": "bed",
      "score": "0.9909400939941406"
    },
    {
      "bbox": "[ 607.98364258  366.91992188  663.56347656  676.80053711]",
      "label": "curtain",
      "score": "0.6649105548858643"
    }
    ...
```


Microsoft

{
"tags": [
{
"bbox": "[168.90582275 493.13183594 299.44207764 609.17419434]",
"label": "pillow",
5234 707.32818604]",
75 502.35699463]",
752 648.09082031]",
3154 832.]",
7656 676.80053711]",



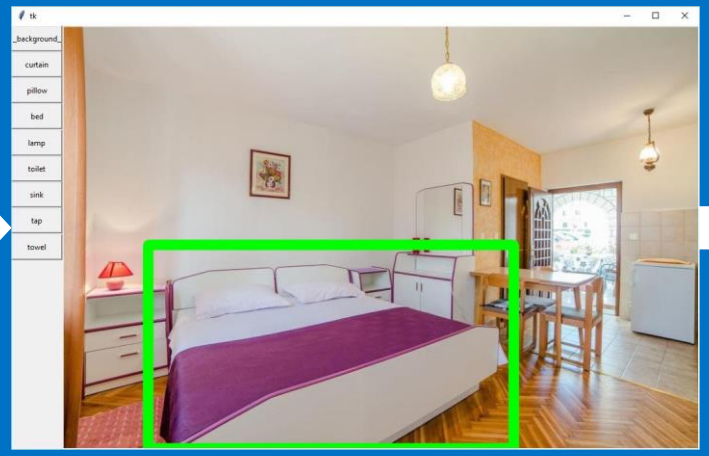
Selected p
of hotel t
be sent
tailored

o evaluate
object
ecide what
ure it is

Preparing a dataset for training

- Pictures for POC dataset were provided by Hotailors
- Tagging was done with Python scripts from CNTK 2.1 that we improved a bit: https://github.com/karolzak/CNTK-Hotel-pictures-classificator/tree/master/Detection/ImageTaggingTool

Draw rectangles
on objects with C1



Assign labels
to each rectangle with C2

When in doubt –
View and verify rectangles with C3

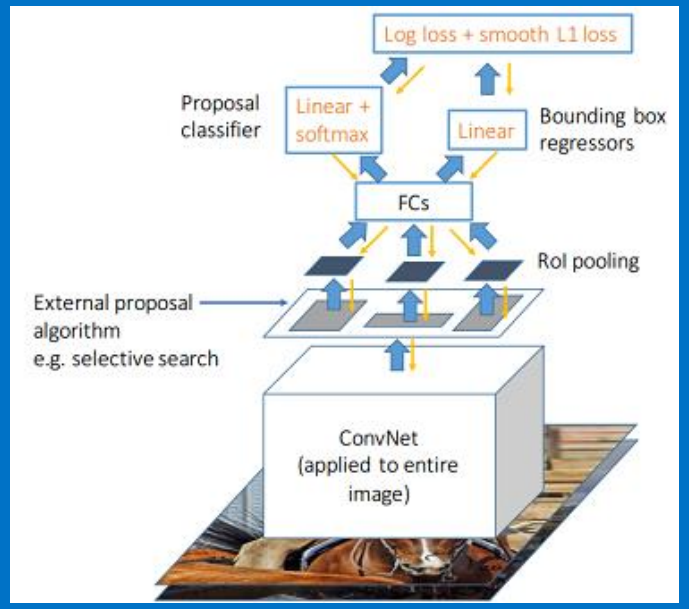
- There's another more advanced tool if you prefer: https://github.com/Microsoft/VoTT

Microsoft

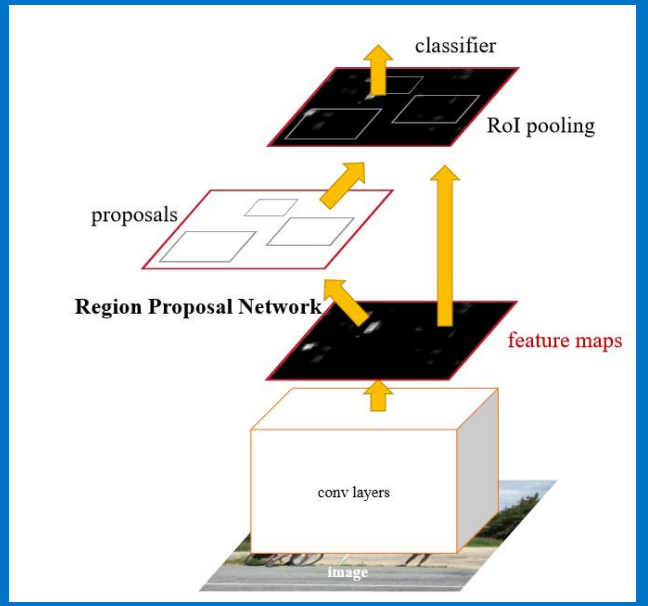
Training the model with CNTK

- microsoft.com/en-us/cognitive-toolkit & github.com/Microsoft/CNTK/
- We first started with FastRCNN but after some research and testing we decided (because of significantly better performance) to use FasterRCNN to train our model for object detection and classification
- FastRCNN and FasterRCNN are two types of Region-based Convolutional Neural Networks used for object detection

Fast-RCNN:



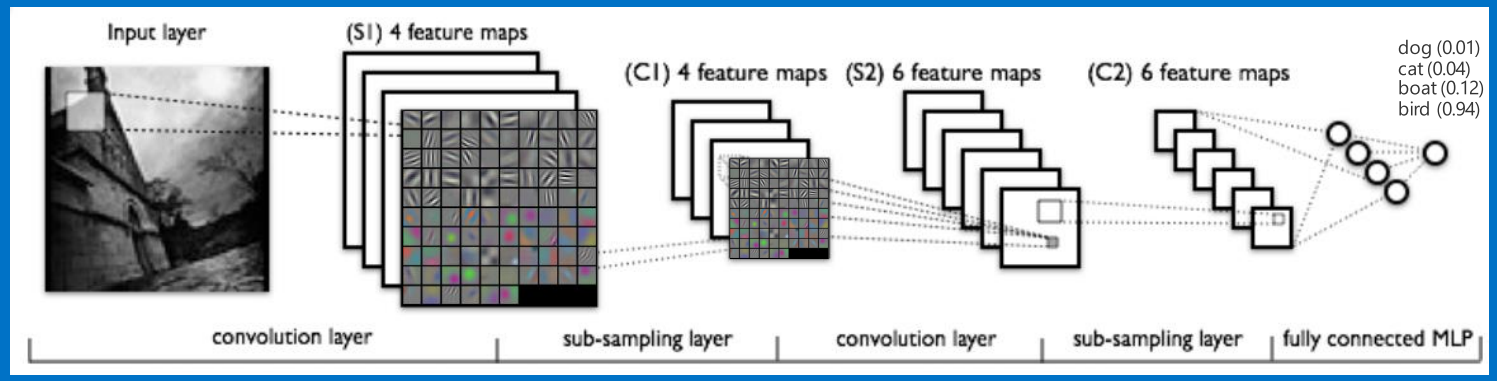
Faster-RCNN:

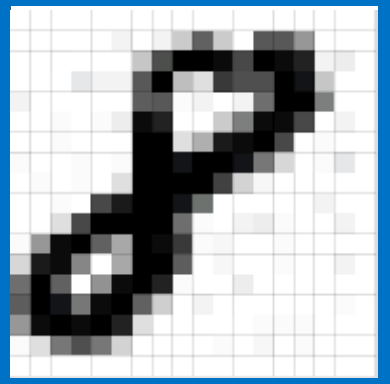


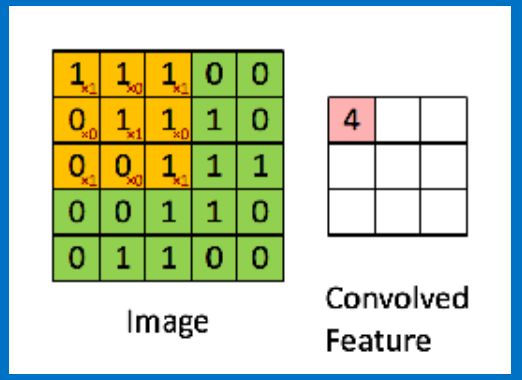
Microsoft

Convolutional Neural Networks

- Convolutional layers apply a convolution operation to the input (using different filters and weights), passing the result as an input for next layer





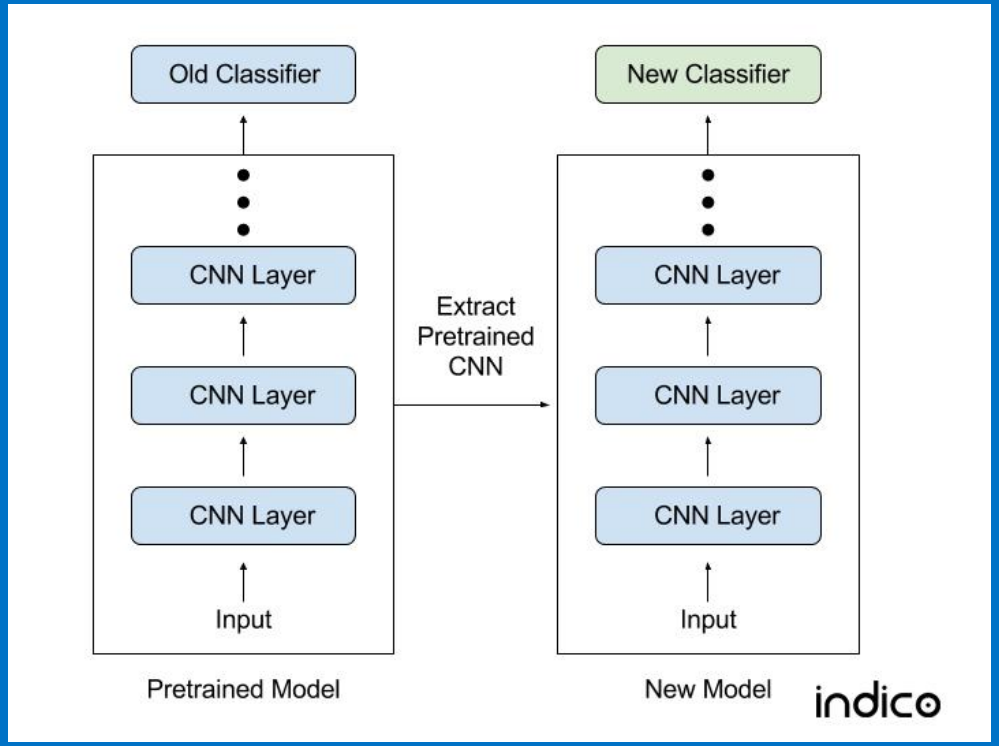


- Every CNN at it's beginning is initialized with random weights and over time and training iterations on our training dataset, it finds the best set of parameters
- In order to teach CNN to recognize shapes and edges you need to train it on a really huge (and LABELED!!) dataset
- Luckly there's a technic called **Transfer Learning!**

Microsoft

Transfer Learning

- Transfer Learning allows you to extract CNN layers with freezed filters and weights from existing (and much more advanced) model and use it to create new model with different classifier
- Popular existing datasets:
  ImageNet (15M images / 22K categories),
  ILSVRC (1.2M / 1K categories) – subset of ImageNet
  ImageNet Large Scale Visual Recognition Competition
  COCO dataset
- Popular (award-winning) models:
  AlexNet (trained on ImageNet dataset),
  VGG, **ResNet**



Microsoft

Results

- It's recommended to train your model using a machine with strong GPU (multiple GPU on multiple machines would be perfect ☺). It's possible to train using CPU but depending on project it may be reeeeally time consuming…
- In our case we used Azure Data Science VMs images with N-series VMs (Nvidia K80 GPUs)

Our dataset in numbers per class:

object/class name	# of tagged objects in positive/train set	# of tagged objects in test set
sink	46	10
pillow	98	27
toilet	34	7
lamp	69	18
curtain	78	16
towel	30	14
tap	44	9
bed	53	12

```
Evaluating Faster R-CNN model for 20 images.
Number of rois before non-maximum suppression: 512
Number of rois  after non-maximum suppression: 82
AP for              tap = 0.0556
AP for          curtain = 0.7930
AP for             sink = 0.4833
AP for           toilet = 0.7143
AP for             lamp = 0.5185
AP for            towel = 0.0000
AP for           pillow = 0.3241
AP for              bed = 0.9167
Mean AP = 0.4757
```

Microsoft

Bonus: Deploying model as Python API to Azure Web Apps

- Just for our POC purpose we decided to create a simple RESTful API with our model and deploy it to Azure Web Apps
- Flask to create web service
- Default Python for Azure Web Apps is 2.7 or 3.4 and we needed 3.5
- Custom Python 3.5.x extension
- Deployment script with custom Python env setup and installation of dependencies
- Repo: https://github.com/karolzak/CNTK-Python-Web-Service-on-Azure
- OUTCOME: It was quite a fun, but it turned out that Azure Web Apps is not the best place for that kind of tasks ☺
  Containers on VMs with GPU are much better for this type of scenarios

Useful links

- Main project repo: https://github.com/karolzak/CNTK-Hotel-pictures-classificator
- Python REST API with trained model and deployment to Azure Web Apps: https://github.com/karolzak/CNTK-Python-Web-Service-on-Azure
- Visual object Tagging Tool: https://github.com/Microsoft/VoTT
- FastRCNN and FasterRCNN research papers
- CNNs explained
- Azure Data Science VMs
- AML Services and Workbench
- Azure Batch AI

https://www.tech-acceleration.com/cz/

O programu

Svět prochází rychlými změnami a spolu s tím rostou požadavky zákazníků. Změnit tradiční procesy a použít moderní technologie už není luxus ale nezbytnost. Pro mnoho firem je ale zavedení inovací velmi náročné – ať už kvůli nepochopení scénářů použití dané technologie nebo chybějícím zdrojům, případně kompetencím. Mezinárodní program Tech Acceleration je určen společnostem, které plánují inovativní projekty, transformují své podnikání a potřebují expertní technickou pomoc, aby byly úspěšné.

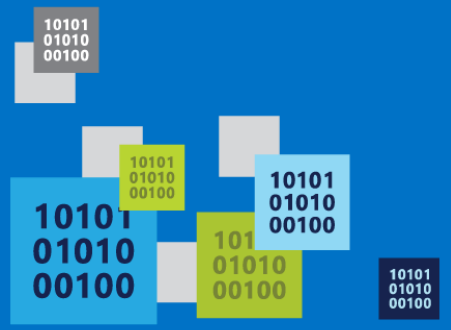
Co nabízíme?

- Setkání nad technologickou vizí. Pomůžeme vám pochopit, zda a jak vám mohu pomoct technologie jako machine learning, cognitive services nebo mikroslužby.
- Podpora při implementaci pilotního projektu. Spolu s vaším technickým týmem navrhou naši specialisté architekturu, napíšou zdrojový kód a rozeberou možnosti dalšího rozvoje.

Thank you!
Q&A

Karol Żak
Software Engineer
Commercial Software Engineering
Microsoft | Central Eastern Europe

karol.zak@microsoft.com
@KarolZak13
karolzak



Microsoft