# Detection of vegetation alongside of the roads using convolution neural network

Toni Antunović Mladen Kukulić Maite Zabala Ugarteburu Una Pale

### Project task

• In given pictures detect which pixels represent vegetation



#### • Difficulties:

- cars, people, traffic signs etc.
- different colors of vegetation
  - (season change, shadows, grass, trees...)

#### Data to analyze

• Pairs of a picture and a mask



# The tool: Caffe

- Processing a large number of pictures and data
  > deep CNN
- CAFFE deep learning framework
- Set of input files
  - Network definition .prototxt
  - Training parameters solver.prototxt
  - Training and validation data set (LMDB)



### Preprocessing

- Resizing images
  - 1080\*1920 px -> 270\*480 px
- Cutout from the image 70\*70 px => "tile"
  - (x,y) pixel is central pixel of a tile
  - Tile size DX\*DY
  - Problem:
    - Pixels which are too close to the edges
  - Solution:
    - Enlarging the picture
    - By adding mirrored strips of on the edges of picture



DX

DY

#### Preprocessing



Enlarged picture

# Preprocessing

- Additional steps
  - Scaling pixels values from [0,255] to [0,1]
  - Substracting the mean value
- Final steps
  - tile paired with (x,y) pixel from corresponding mask image
    - Pairs written in .txt file -> LMDB input file for network crops\_train/3\_1.jpeg 0 crops\_train/3\_2.jpeg 1
    - 50% of positive and 50% od negative examples
      examples shuffled

# Postprocessing

- Test images
  - Crop tiles for every pixel on picture
    - Row by row
    - Put it into network
    - Get result
  - Using masks and ROC curves find the optimum percentage
    - Above which the pixels on output image is set to 1
  - Output is bw image
    - Compare to mask
    - Calculate accuracy
  - Repeat for every image in test set
    - Average accuracy

# Results

- Several networks constructed
  - For all of them same problem occured
  - During training
    - Loss function is not descending continuously but oscillates
  - In test stage
    - Outputs of the network are the same for every tile given

# Results

- Things done
  - Different networks tried
  - Scaling image values from [0,255] to [0,1]
  - Substracting mean
  - Slowing down learning rate
  - Shuffeling training dana set
- Future task
  - Continue with experimenting

# The end!

