



NVIDIA DIGITS

DEEP LEARNING GPU TRAINING SYSTEM

BRUNO AUGUSTO DORTA MARQUES

Resumo

1. O que é DIGITS
2. Como utilizar o DIGITS
3. Redes de Convolução
4. Treinando um classificador de dígitos
5. Treinando uma rede de segmentação de imagens

Curso disponível em:

<http://www2.ic.uff.br/~gpu/learn-gpu-computing/deep-learning>

Exemplos práticos de Deep Learning



Classificação de Imagens,
Detecção e Localização de
Objetos,
Reconhecimento de Ações,
Descrição de Cenas



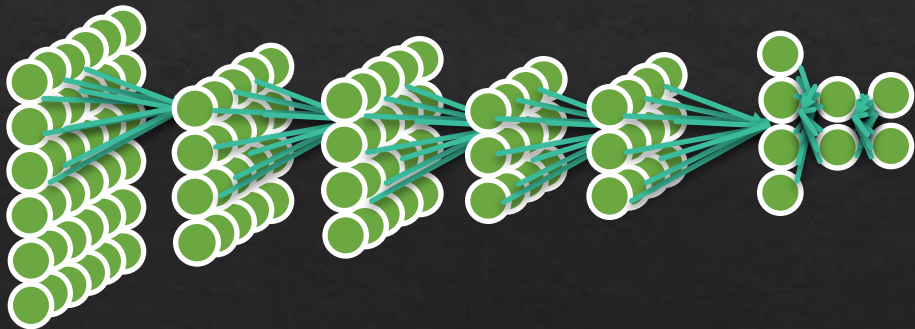
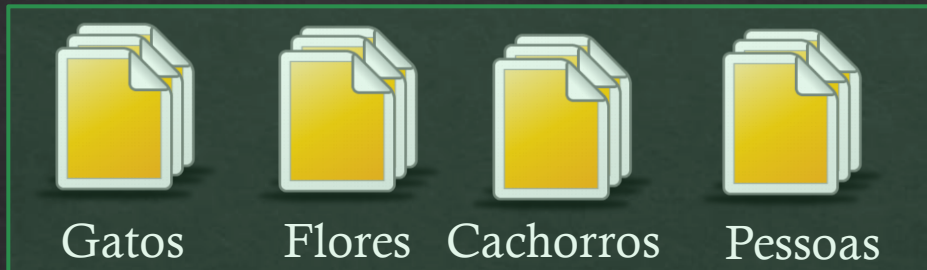
Detecção de Pedestres,
Reconhecimento de Sinais de
Trânsito



Reconhecimento de Fala,
Tradução de Fala,
Processamento de Linguagem
Natural

Classificação de imagens

Treinamento



Processo de treinamento :

- ◇ Escolha de uma Rede Neural Profunda
- ◇ Fornecer como entrada 200 milhões de imagens de treinamento pertencentes a 5000 categorias
- ◇ O Processo pode levar dias
- ◇ Se acurácia estiver baixa podemos:
 - ◇ Modificar a arquitetura da Rede
 - ◇ Modificar o conjunto de treinamento
 - ◇ Atualizar os parâmetros de treinamento.

DIGITS

- ◇ Ferramenta de Visualização para Treinamento de Redes Neurais Profundas
- ◇ Permite utilizar redes padrões, importar novas redes, construir sua própria rede
- ◇ Importar dados de treinamento e criar base de dados a partir do disco rígido
- ◇ Monitorar o processo de treinamento de múltiplas redes

DIGITS

- ◇ Disponível em:
 - ◇ developer.nvidia.com/digits
 - ◇ github.com/NVIDIA/DIGITS
- ◇ Gratuito
- ◇ Pode ser utilizado em Linux, Windows, MacOS
- ◇ Suporte a classificação, detecção e segmentação de imagens.
- ◇ Versões Futuras: Mais problemas e formatos de dados

- ◇ Permite utilizar os seguinte Frameworks:
 - ◇ Caffe
 - ◇ Torch
 - ◇ Tensor Flow

Como utilizar DIGITS

- ◇ Iniciando o Server:
 - ◇ 'digits-devserver'
 - ◇ Inicia digits na porta 5000
 - ◇ 'digits-server'
 - ◇ Inicia digits como um serviço web, configurável com nginx

- ◇ Acessando a Interface DIGITS:
 - ◇ Localmente, no navegador: <http://localhost/> ou localhost:5000
 - ◇ Remoto, no navegador: ip:8080 ou ip:5000

Home

2/2 GPUs available

No Jobs Running

Datasets (63)

Models (64)

New Dataset

New Model

Images ▾

Images ▾

Delete



Filter

name	refs	backend	status	elapsed	submitted [▲]
BodyMotion(HevertonBreno)	1	Imdb	Done	11m	Jan 20, 17
BodyMotion(HevertonDiego)4classes	5	Imdb	Done	6m	Jan 20, 17
BodyMotion(HevertonDiego)6classes	1	Imdb	Done	11m	Jan 19, 17
BodyMotion(HevertonDiego)	2	Imdb	Done	24m	Jan 19, 17
Umale_FCN_database_Train_and_Val_smaller_val	2		Done	4m	Jan 9, 17

❖ Criar o Banco de Dados

DIGITS Login Info

Home

2/2 GPUs available

No Jobs Running

Datasets (63) Models (64)

New Dataset New Model

Images Images

Classification
Gradients
Object Detection
Other

name	size	backend	status	created	updated
BodyMotion(HevertonBreno)	1	lmbd	Done		
BodyMotion(HevertonDiego)4classes	5	lmbd	Done	6m	Jan 20, 17
BodyMotion(HevertonDiego)6classes	1	lmbd	Done	11m	Jan 19, 17
BodyMotion(HevertonDiego)	2	lmbd	Done	24m	Jan 19, 17

❖ Configurar Modelo de Rede

New Image Classification Dataset

Image Type **?**
Color

Image size (Width x Height) **?**
256 x 256

Resize Transformation **?**
Squash

See example

Use Image Folder Use Text Files

Training Images **?**
folder or URL

Minimum samples per class **?**
2

Maximum samples per class **?**

% for validation **?**
25

% for testing **?**
0

Separate validation images folder
 Separate test images folder

DB backend
LMDB

New Image Classification Model

Select Dataset **?**

BodyMotion(HevertonB
BodyMotion(HevertonD
BodyMotion(HevertonD
BodyMotion(HevertonD
Human3.6(S11VAL)(S1
Human3.6(S11VAL)(S1

Python Layers **?**

Server-side file **?**

Use client-side file

Solver Options

Training epochs **?**
30

Snapshot interval (in epochs) **?**
1

Validation interval (in epochs) **?**
1

Random seed **?**

Data Transformations

Crop Size **?**
none

Subtract Mean **?**
Image

New Image Classification Dataset

Image Type [?]

Color

Image size (Width x Height) [?]

256

x

256

Resize Transformation [?]

Squash

See example

Use Image Folder

Use Text Files

Training Images [?]

folder or URL

Minimum samples per class [?]

2

Maximum samples per class [?]

% for validation [?]

25

% for testing [?]

0

- Separate validation images folder
- Separate test images folder

DB backend

LMDB

Image Encoding [?]

PNG (lossless)

New Image Classification Dataset

Use Image Folder Use Text Files

Image Type ?
Color

Image size (Width x Height) ?
256 x 256

Resize Transformation ?
Squash

[See example](#)

Training Images ?
folder or URL

Minimum samples per class ?
2

Maximum samples per class ?

% for validation ?
25

% for testing ?
0

Separate validation images folder

Separate test images folder

DB backend
LMDB

Image Encoding ?
PNG (lossless)

Propriedades das imagens
Operação de escala é aplicada a
todas as imagens

Squash



Crop



Fill



New Image Classification Dataset

Image Type ?

Color

Image size (Width x Height) ?

256 x 256

Resize Transformation ?

Squash

[See example](#)

Use Image Folder **Use Text Files**

Training Images ?

folder or URL

Minimum samples per class ?

2

Maximum samples per class ?

% for validation ?

25

% for testing ?

0

Separate validation images folder

Separate test images folder

DB backend

LMDB

Image Encoding ?

PNG (lossless)

Use Image Folder Use Text Files

Set	Text file ?	Image folder (optional) ?
	<input type="checkbox"/> Use local paths on server ?	
Training	Browse...	
<input checked="" type="checkbox"/> Validation	Browse...	
<input type="checkbox"/> Test	Browse...	

Shuffle lines ?

Yes

Labels ?

[Browse...](#)

Use Image Folder Use Text Files

Training Images ?

folder or URL

Minimum samples per class ?

2

Maximum samples per class ?

% for validation ?

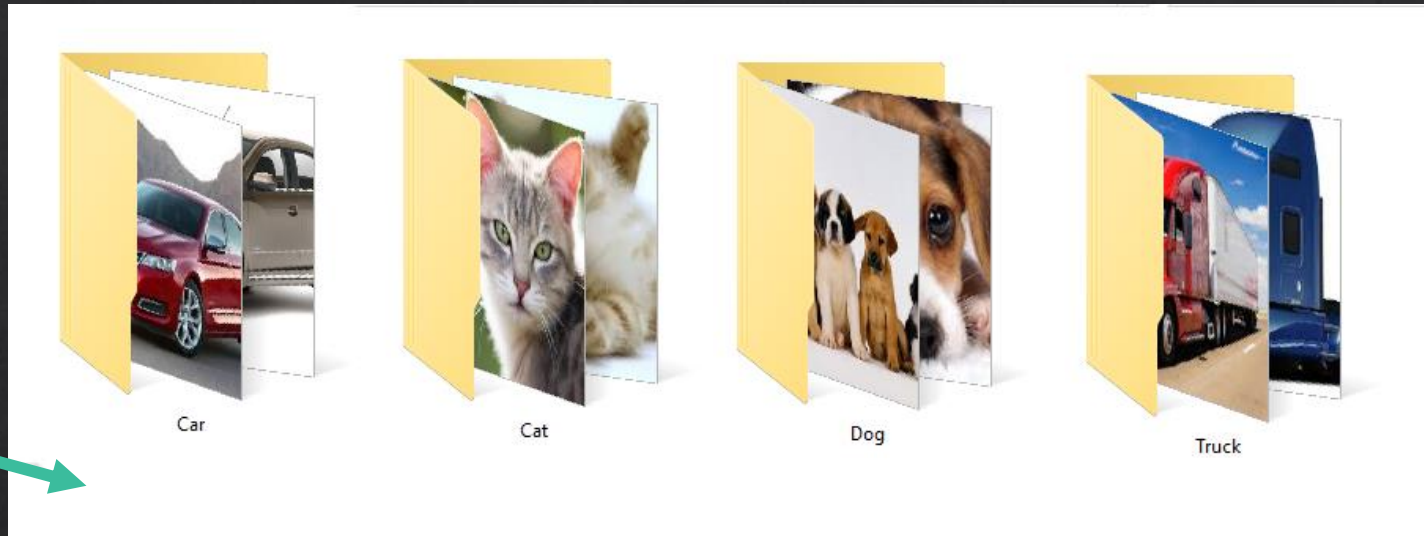
25

% for testing ?

0

Separate validation images folder

Separate test images folder



Diretório contendo pastas com as imagens de treinamento.

Cada diretório deve conter o nome da classe ao qual a imagem pertence.

Use Image Folder

Use Text Files

Training Images ?

/home/bmarques/dataset03/train/

Minimum samples per class ?

2

Maximum samples per class ?

% for validation ?

25

% for testing ?

0

Separate validation images folder

Validation Images

/home/bmarques/dataset03/val/

Minimum samples per class ?

Maximum samples per class ?

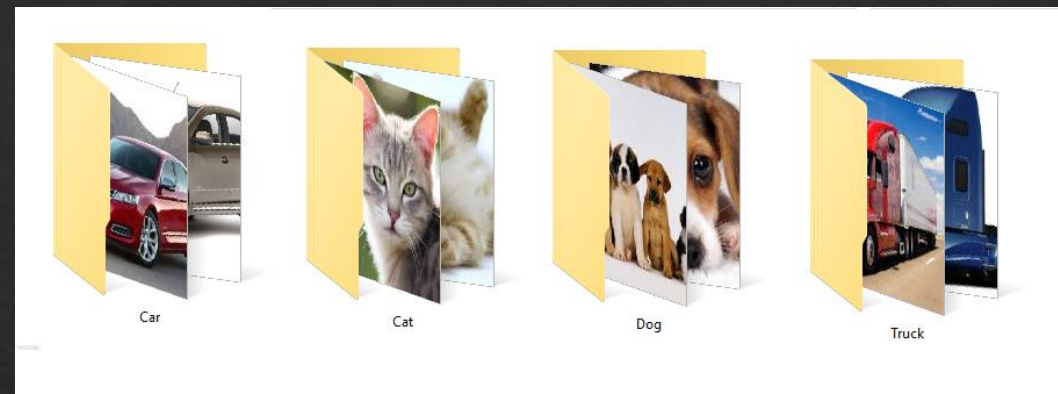
Separate test images folder

Test Images

/home/bmarques/dataset03/test/

Minimum samples per class ?

Maximum samples per class ?



/dataset/train/



/dataset/val/



/dataset/test/

Use Image Folder Use Text Files

Set	Text file ?	Image folder (optional) ?
	<input type="checkbox"/> Use local paths on server ?	
Training	<input type="button" value="Browse..."/> <input type="text"/>	<input type="text"/>
<input checked="" type="checkbox"/> Validation	<input type="button" value="Browse..."/> <input type="text"/>	<input type="text"/>
<input type="checkbox"/> Test	<input type="button" value="Browse..."/> <input type="text"/>	<input type="text"/>

Shuffle lines ?

Labels ?

Train.txt

```
/home/bmarques/dataset03/000114.jpg 0  
/home/bmarques/dataset03/009936.jpg 0  
/home/bmarques/dataset03/000060.jpg 1  
/home/bmarques/dataset03/011383.jpg 0  
/home/bmarques/dataset03/000473.jpg 1  
/home/bmarques/dataset03/000350.jpg 3
```

Val.txt

```
/home/bmarques/dataset03/010228.jpg 0  
/home/bmarques/dataset03/010457.jpg 2  
/home/bmarques/dataset03/001449.jpg 2  
/home/bmarques/dataset03/011085.jpg 0  
/home/bmarques/dataset03/010365.jpg 3  
/home/bmarques/dataset03/010170.jpg 1  
/home/bmarques/dataset03/010849.jpg 0  
/home/bmarques/dataset03/000468.jpg 2  
/home/bmarques/dataset03/011208.jpg 0
```

Job Information

Job Directory
/usr/share/digits/digits/jobs/20160911-161301-7ad2

Image Dimensions
256x256 (Width x Height)

Image Type
Color

Resize Transformation
Squash

DB Backend
Imdb

Image Encoding
jpg

DB Compression
none

Dataset size
267 MB

Job Status Done

- Initialized at Sep 11 2016, 04:13:01 PM (1 second)
- Running at Sep 11 2016, 04:13:02 PM (1 minute, 28 seconds)
- Done at Sep 11 2016, 04:14:31 PM (Total - 1 minute, 29 seconds)

Create DB (train) Done ▾

Create DB (val) Done ▾

Create DB (test) Done ▾

Status do Job Atual:

- Estimativa de tempo.
- Hora de início.
- Barra de progresso.

Create DB (train)

Input File (before shuffling)
/home/badmarques/data/npdi/Frames/configfiles/train.txt

DB Creation log file
create_train_db.log

DB Entries
13382

Category	Image Count
Category 1	8300
Category 2	5100

Related jobs

Image Classification Model

NPDI_Original_GoogleNet_Pretrained_03
Done ▾

Notes

None ↗

Informações das classes

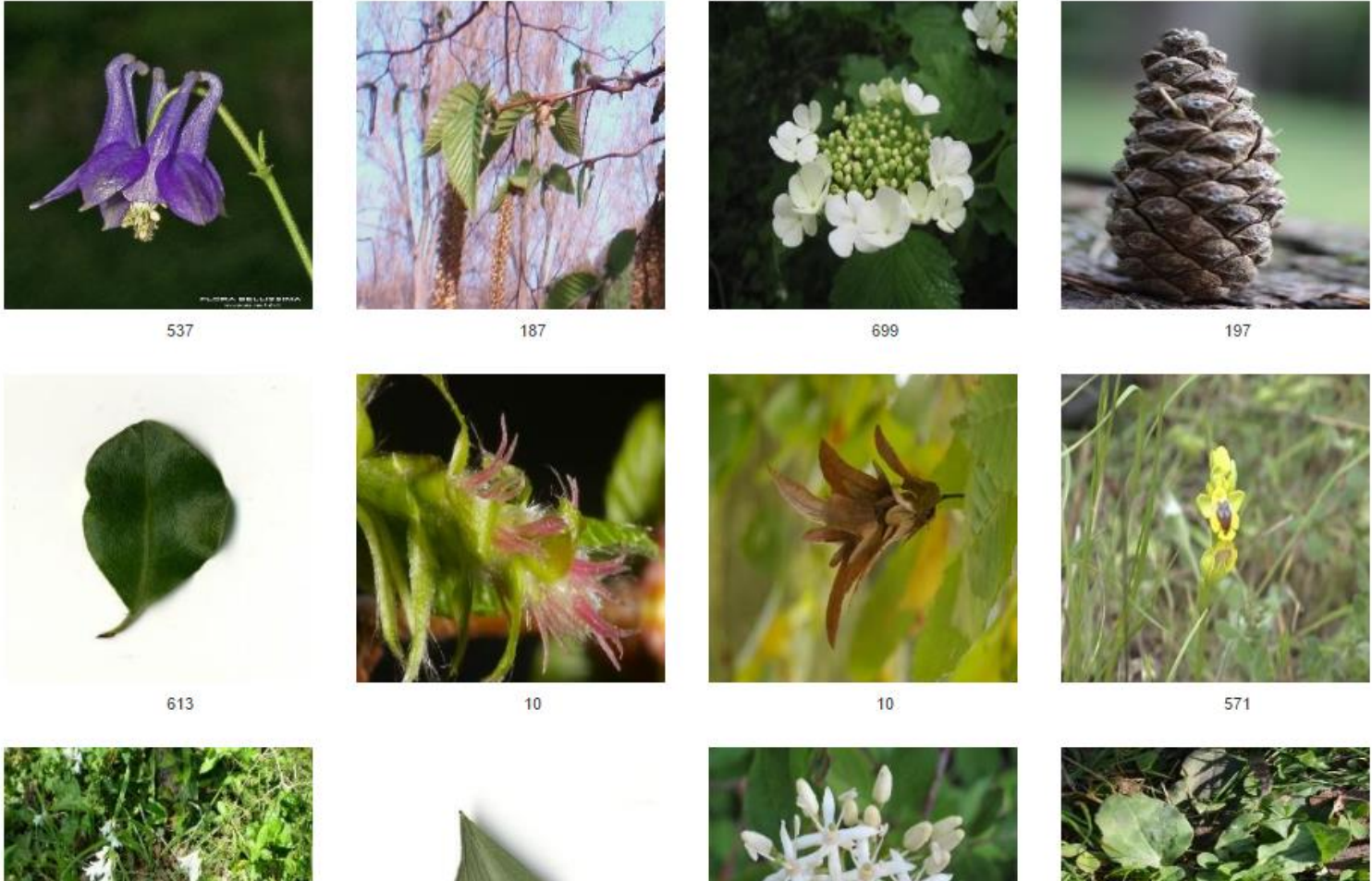
- Quantidade de amostras por classe

Explorar Base de Dados

946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977
978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999

Items per page: 10 - 25 - 50 - **100**

« 0 1 2 3 4 5 ... 734 »



537 187 699 197

613 10 10 571

- ◇ Exibir todas as Imagens e seu rótulo
- ◇ Filtrar por Classe

Modelos de Rede

DIGITS bruno (Logout) Info

Home 2/2 GPUs available

No Jobs Running

Datasets (18) **Models (15)**

New Dataset Images **New Model**

- Images
- Classification
- Object Detection
- Other

Delete

Filter

name	framework	status	elapsed	start
FCN Rhcmelo Thermal U	caffe	Aborted	10d	Dec 19, 16
faces_GoogLeNet_Igomes_ex1	caffe	Done	1h	Dec 8, 16
Rhcmelo FCN finetunning alpha=fixed (1e-12) augmented Mirror_batch4	caffe	Aborted	2d	Dec 5, 16
Rhcmelo FCN finetunning alpha=fixed (1e-12) augmented Mirror_batch4	caffe	Aborted	2d	Dec 5, 16
Plants_Aug_GoogleNet_05	caffe	Done	18h	Oct 18, 16
Plants_Original_GoogleNet_04	caffe	Done	3h	Oct 17, 16
NPDI_Aug_GoogleNet_07_0001	caffe	Done	3h	Oct 13, 16
NPDI_Aug_GoogleNet_06_0001	caffe	Done	1d	Sep 28, 16
Plants_Original_GoogleNet_03	caffe	Done	6h	Sep 20, 16
Plants_Original_GoogleNet_02	caffe	Done	3h	Sep 19, 16
NPDI_Original_GoogleNet_07	caffe	Done	37m	Sep 19, 16
NPDI_Original_GoogleNet_06	caffe	Done	38m	Sep 19, 16
NPDI_Original_GoogleNet_Pretrained_03	caffe	Done	21m	Sep 11, 16
Plants_Original_GoogleNet_Pretrained	caffe	Done	4h	Sep 11, 16
isbi original	caffe	Aborted	19h	Sep 8, 16

◇ Criar um novo modelo

◇ Escolher modelo treinado anteriormente

Configurando um modelo

New Image Classification Model

Select Dataset

- Faces_Igomes
- Plants_Aug_FullLayers
- Plants_Original_Full
- NPDI_Aug_FullLayer_Balanced
- NPDI_AugFull_Balanced
- NPDI_Original_Balanced

Python Layers

Server-side file

Use client-side file

Solver Options

Training epochs

30

Snapshot interval (in epochs)

1

Validation interval (in epochs)

1

Random seed

[none]

Batch size multiples allowed

[network defaults]

Batch Accumulation

Solver type

Stochastic gradient descent (SGD)

Base Learning Rate multiples allowed

0.01

Show advanced learning rate options

Data Transformations

Crop Size

none

Subtract Mean

Image

◇ Selecionar uma base de dados.

- ◇ Escolher parâmetros da rede:
- ◇ Épocas de treinamento
 - ◇ Quantidade de épocas para criar snapshots e validação
 - ◇ Tamanho do Batch
 - ◇ Tipo de Solver
 - ◇ Learning Rate

Escolha da Arquitetura de Rede

Standard Networks Previous Networks Custom Network

Caffe Torch

Network	Details	Intended image size
<input type="radio"/> LeNet	Original paper [1998]	28x28 (gray)
<input type="radio"/> AlexNet	Original paper [2012]	256x256
<input type="radio"/> GoogLeNet	Original paper [2014]	256x256

Use this many GPUs (next available)

1

or

Select which GPU[s] you would like to use ⓘ

#0 - GeForce GTX TITAN X (11.9 GB memory)
#1 - GeForce GTX TITAN X (11.9 GB memory)

Model Name ⓘ

Create

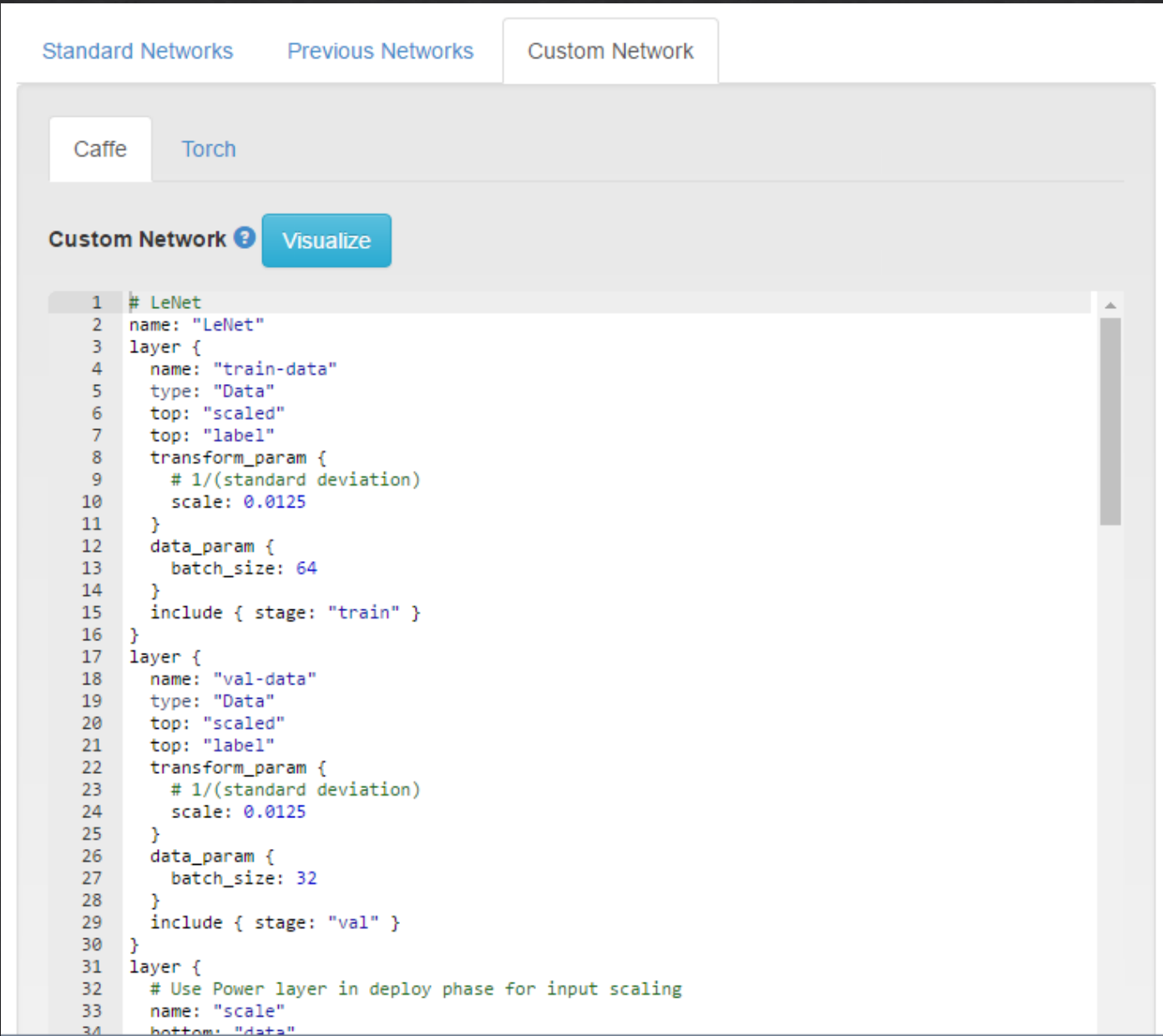
◇ Podemos escolher uma das redes padrões:

◇ LeNet

◇ AlexNet

◇ GoogLeNet

Escolha da Arquitetura de Rede

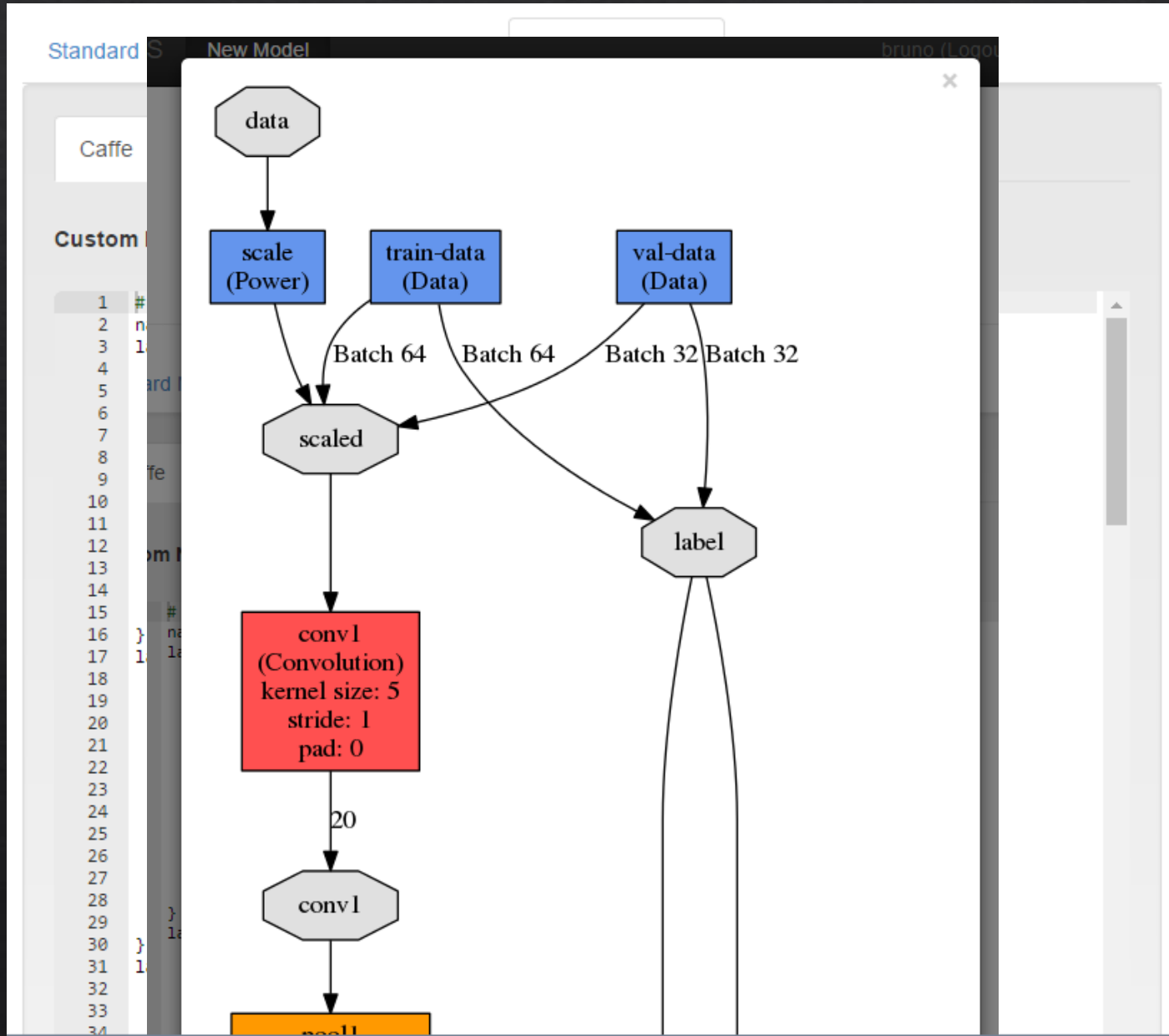


The screenshot shows the Caffe web interface with the 'Custom Network' tab selected. The 'Caffe' framework is chosen. A 'Visualize' button is visible next to the network name. The network configuration is as follows:

```
1 # LeNet
2 name: "LeNet"
3 layer {
4   name: "train-data"
5   type: "Data"
6   top: "scaled"
7   top: "label"
8   transform_param {
9     # 1/(standard deviation)
10    scale: 0.0125
11  }
12  data_param {
13    batch_size: 64
14  }
15  include { stage: "train" }
16 }
17 layer {
18   name: "val-data"
19   type: "Data"
20   top: "scaled"
21   top: "label"
22   transform_param {
23     # 1/(standard deviation)
24    scale: 0.0125
25  }
26  data_param {
27    batch_size: 32
28  }
29  include { stage: "val" }
30 }
31 layer {
32   # Use Power layer in deploy phase for input scaling
33   name: "scale"
34   bottom: "data"
```

- ◇ Podemos customizar ou criar uma nova arquitetura
- ◇ Caffe, Torch, Tensorflow Framework
- ◇ É possível visualizar a rede de forma gráfica.

Escolha da Arquitetura de Rede



- ◇ Podemos customizar ou criar uma nova arquitetura
- ◇ Caffe, Torch, Tensorflow Framework
- ◇ É possível visualizar a rede de forma gráfica.

Caffe

- ◇ Framework de Deep Learning desenvolvido pela Berkeley AI Research.
- ◇ Open Source
- ◇ Suporta CPU e GPU

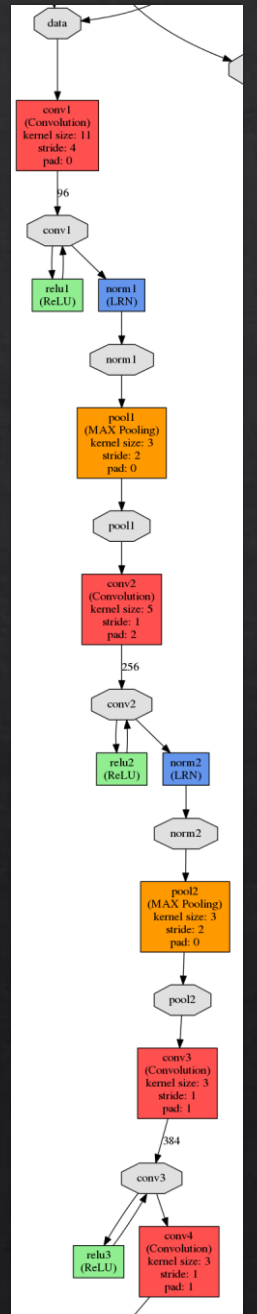
- ◇ Disponível em :
- ◇ BVLC Caffe: <https://github.com/BVLC/caffe>
- ◇ Nvidia Caffe: <https://github.com/NVIDIA/caffe>

Jia, Yangqing, et al. "Caffe: Convolutional architecture for fast feature embedding." *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014.

Definição de rede

- ◆ Utiliza o prototxt para definição de uma rede
- ◆ Definição por Camadas, Modular:

```
layer {  
  name: "conv1"  
  type: "Convolution"  
  bottom: "data"  
  top: "conv1"  
  convolution_param {  
    num_output: 96  
    kernel_size: 11  
    stride: 4  
  }  
}
```



Redes de Convolução

- ◇ Redes que operam sobre imagens
- ◇ Composta principalmente pelas camadas:
 - ◇ Camada de Convolução - Convolution Layer
 - ◇ Camada Não Linear (ReLU)
 - ◇ Camada de Sub Amostragem - Pooling Layer
 - ◇ Camada Totalmente Conectada – Full Connected Layer

Camada de Dados

- ◇ Determina o tipo de dado para cada etapa:
- ◇ Treinamento, validação e treino.
- ◇ Não é necessário no DIGITS.

```
layer {  
  name: "train-data"  
  type: "Data"  
  bottom: "data"  
  top: "data"  
  top: "label"  
  
  data_params {  
    batch_size: 64  
  }  
  include { stage: "train" }  
}
```

Camada de Convolução

- ◆ É composta por Feature Maps.
 - ◆ O Feature Map é resultante da operação de convolução da imagem por filtros aprendidos pela rede.
- ◆ Parâmetros importantes de uma camada de convolução:
 - ◆ Profundidade : Quantidade de filtros utilizados
 - ◆ Passo (Stride) : número de pixels que o filtro irá utilizar para deslizar sobre a imagem.
 - ◆ Zero-padding : adição de zeros ao redor da borda da imagem.



Camada de Convolução

- ◆ É composta por Feature Maps.
 - ◆ O Feature Map é resultante da operação de convolução da imagem por filtros aprendidos pela rede.
- ◆ Parâmetros importantes de uma camada de convolução:
 - ◆ Profundidade : Quantidade de filtros utilizados
 - ◆ Passo (Stride) : número de pixels que o filtro irá utilizar para deslizar sobre a imagem.
 - ◆ Zero-padding : adição de zeros ao redor da borda da imagem.

```
layer {  
  name: "conv1"  
  type: "Convolution"  
  bottom: "data"  
  top: "conv1"  
  convolution_param {  
    num_output: 96  
    kernel_size: 11  
    stride: 4 }  
}
```

ReLu

- ◇ Adiciona não linearidade a rede.
- ◇ Tradicionalmente aplicada após uma camada de convolução
- ◇ $\text{Max}(\text{zero}, \text{output})$

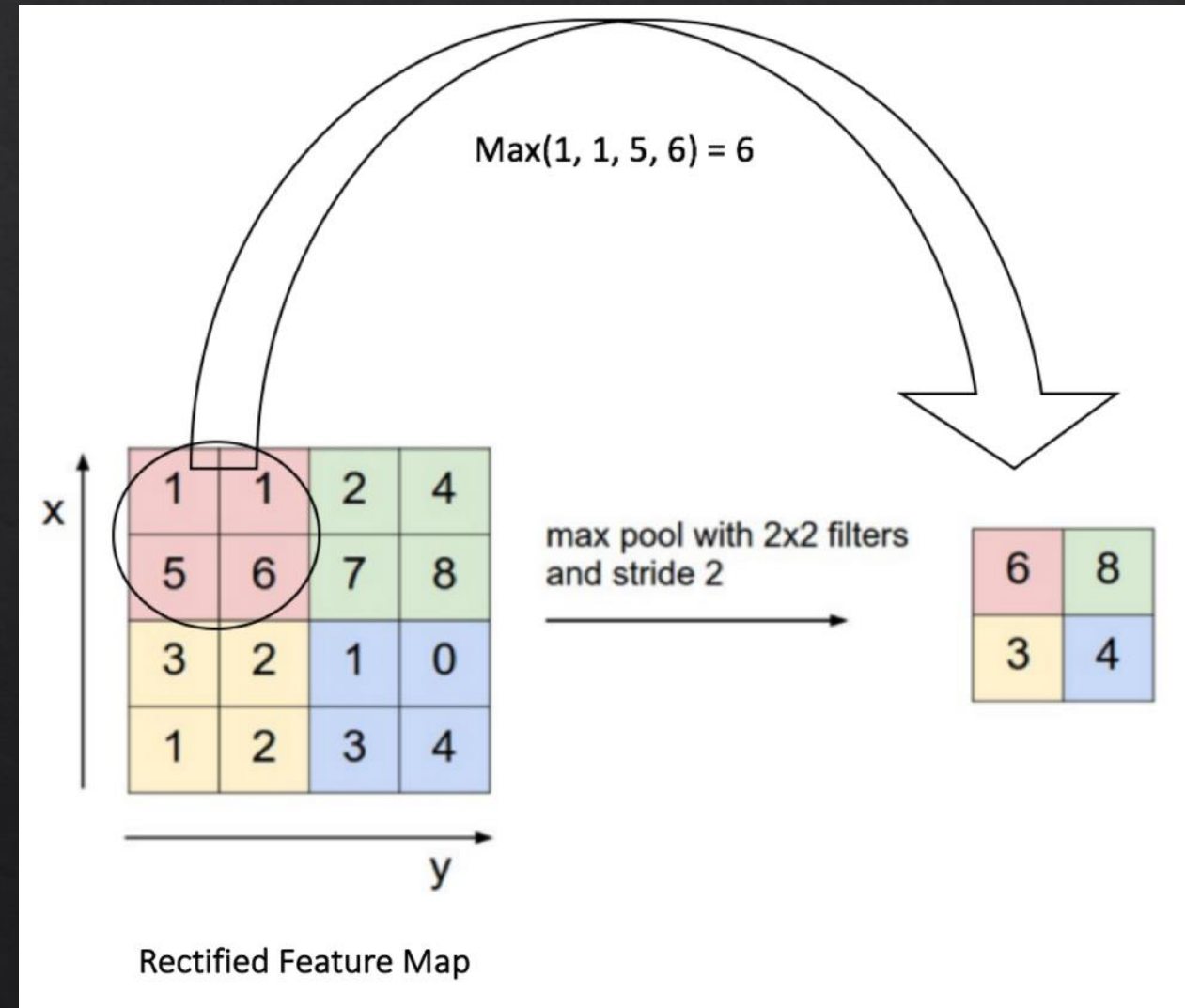
- ◇ A função é aplicada a cada pixel da entrada.

- ◇ Outras funções que podem ser utilizadas:
 - ◇ **tanh** or **sigmoid**

```
layer {  
  name: "relu1"  
  type: "ReLU"  
  bottom: "conv1"  
  top: "conv1"  
}
```

Camada de Pooling

- ◇ Reduz a dimensionalidade da entrada, porém preserva a informação
- ◇ Deslizamos uma janela sobre a imagem aplicando uma função de pooling:
- ◇ Tipos de pooling:
 - ◇ **Max Pooling**
 - ◇ Average Pooling
 - ◇ Sum



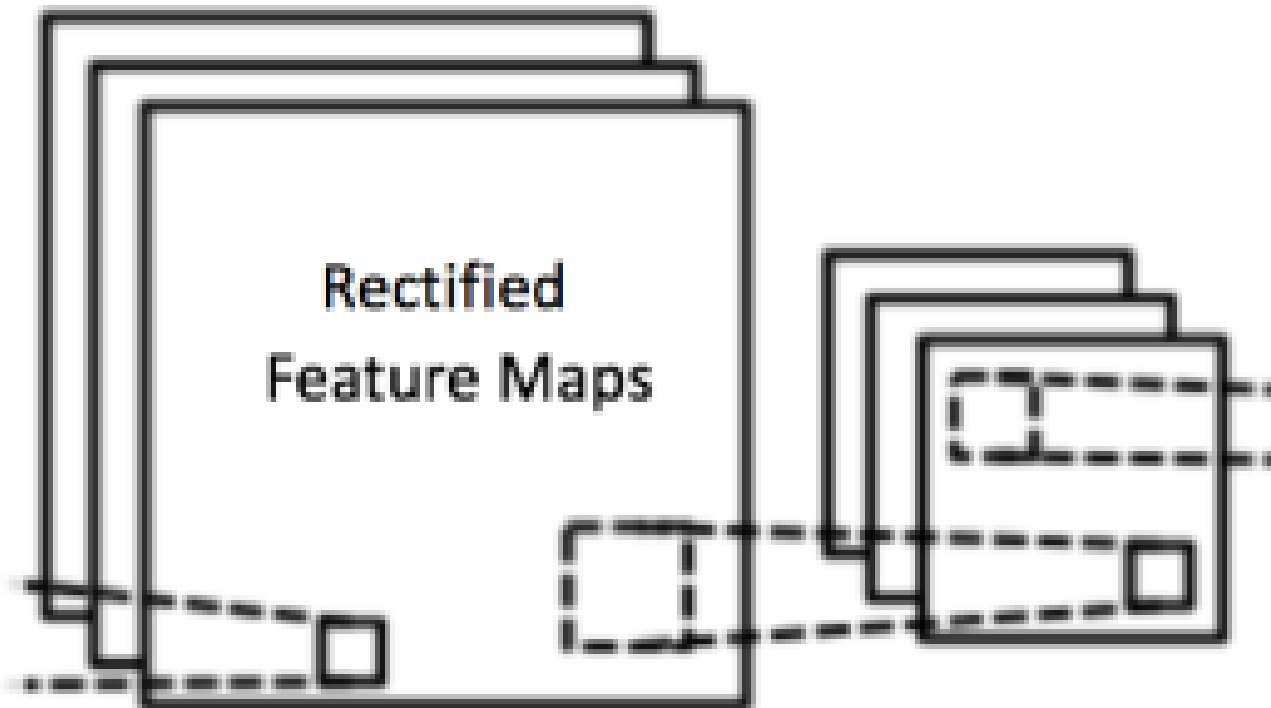
Camada de Pooling

- ◇ Reduz a dimensionalidade da entrada, porém preserva a informação
- ◇ Deslizamos uma janela sobre a imagem aplicando uma função de pooling:
- ◇ Tipos de pooling:
 - ◇ **Max Pooling**
 - ◇ Average Pooling
 - ◇ Sum

```
layer {  
  name: "pool1"  
  type: "Polling"  
  bottom: "conv1"  
  top: " pool1"  
  pooling_param {  
    pool: MAX  
    kernel_size: 3  
    stride: 2  
  }  
}
```

Convolution
using 3 filters
+ ReLU

Pooling applied
separately on each
feature map

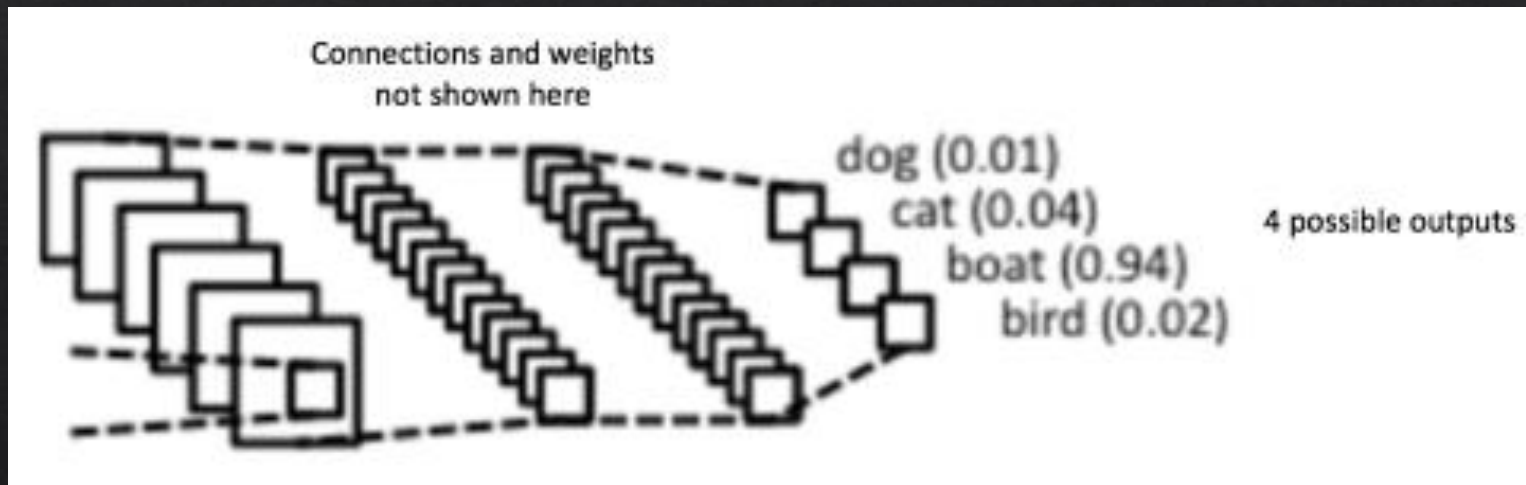


Importância do Pooling

- ◇ Diminuir a carga computacional, facilitar o treinamento
- ◇ Reduz o número de parâmetros, ajudando a controlar o overfitting
- ◇ A rede se torna invariante a pequenas transformações, distorções e translações da imagem original

Camada Totalmente Conectada

- ◇ Multilayer Perceptron com um classificador Softmax
- ◇ Classifica uma imagem de acordo com a combinação de features da ultima camada conectada a ela.
- ◇ Gera uma saída normalizada com a probabilidade da imagem para cada classe

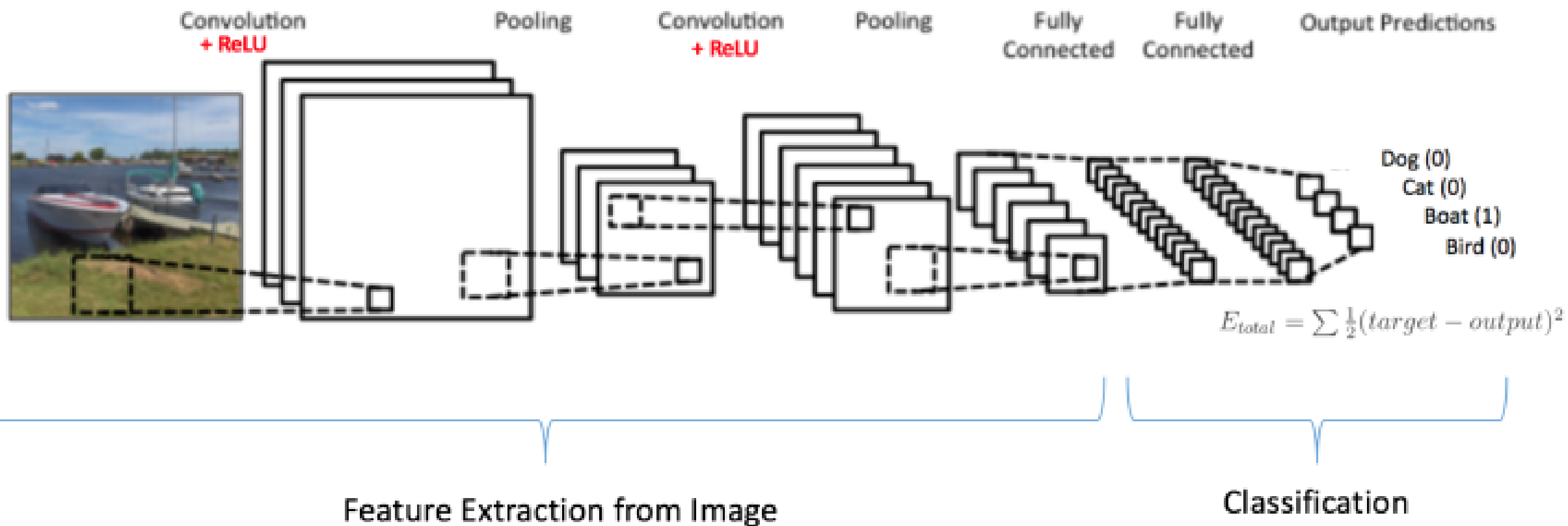


Camada Totalmente Conectada

- ◇ `num_output` = Quantidade de classes
- ◇ Digits é capaz de estimar o `num_output` através da quantidade de classes do dataset

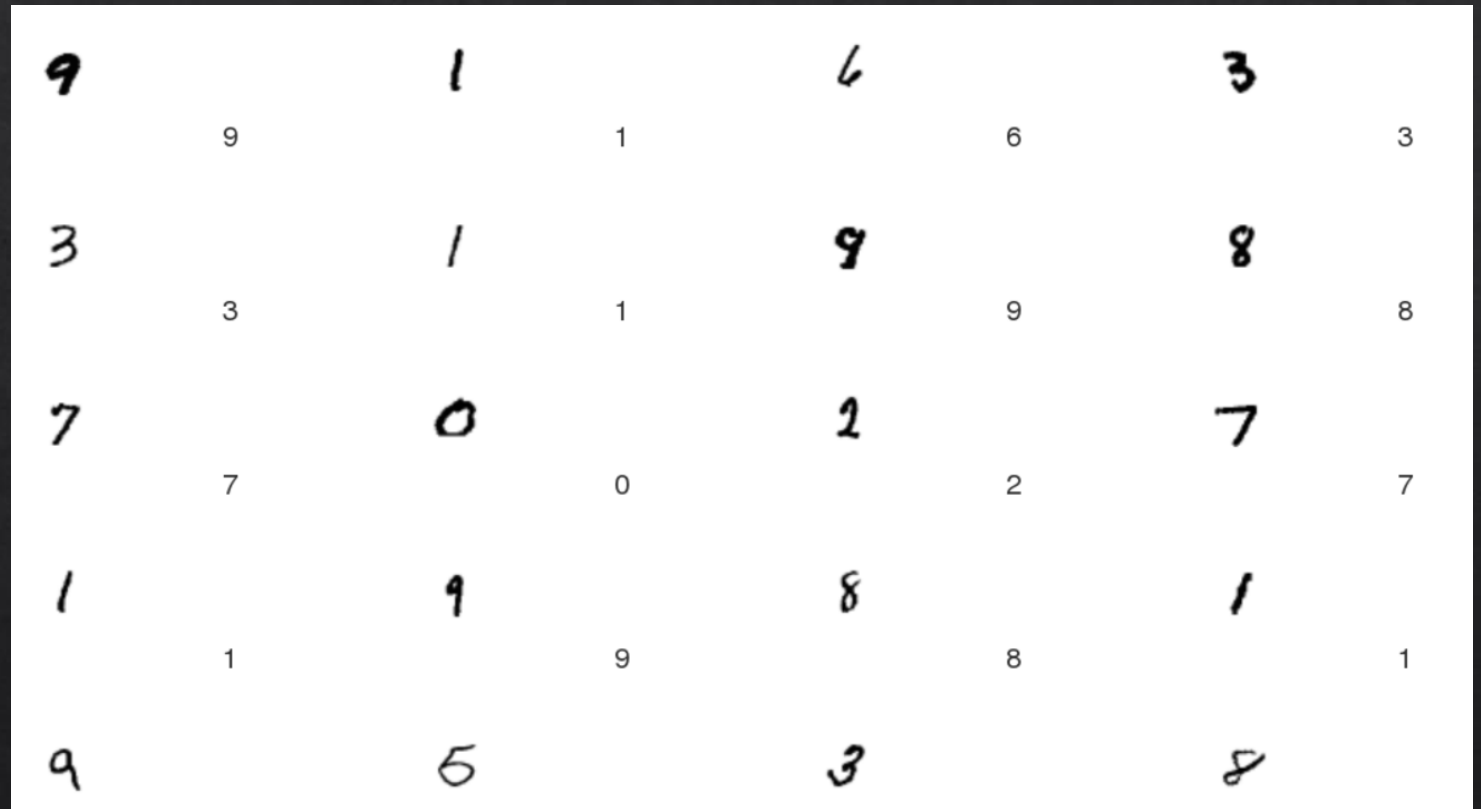
```
layer {
  name: "fc"
  type: "InnerProduct"
  bottom: "pool7"
  top: " fc"
  inner_product_param {
    #num_output: 1000
  }
}
Layer {
  name: "loss"
  type: "SoftmaxWithLoss"
  bottom: "fc"
  top: " loss"
}
```

LeNet



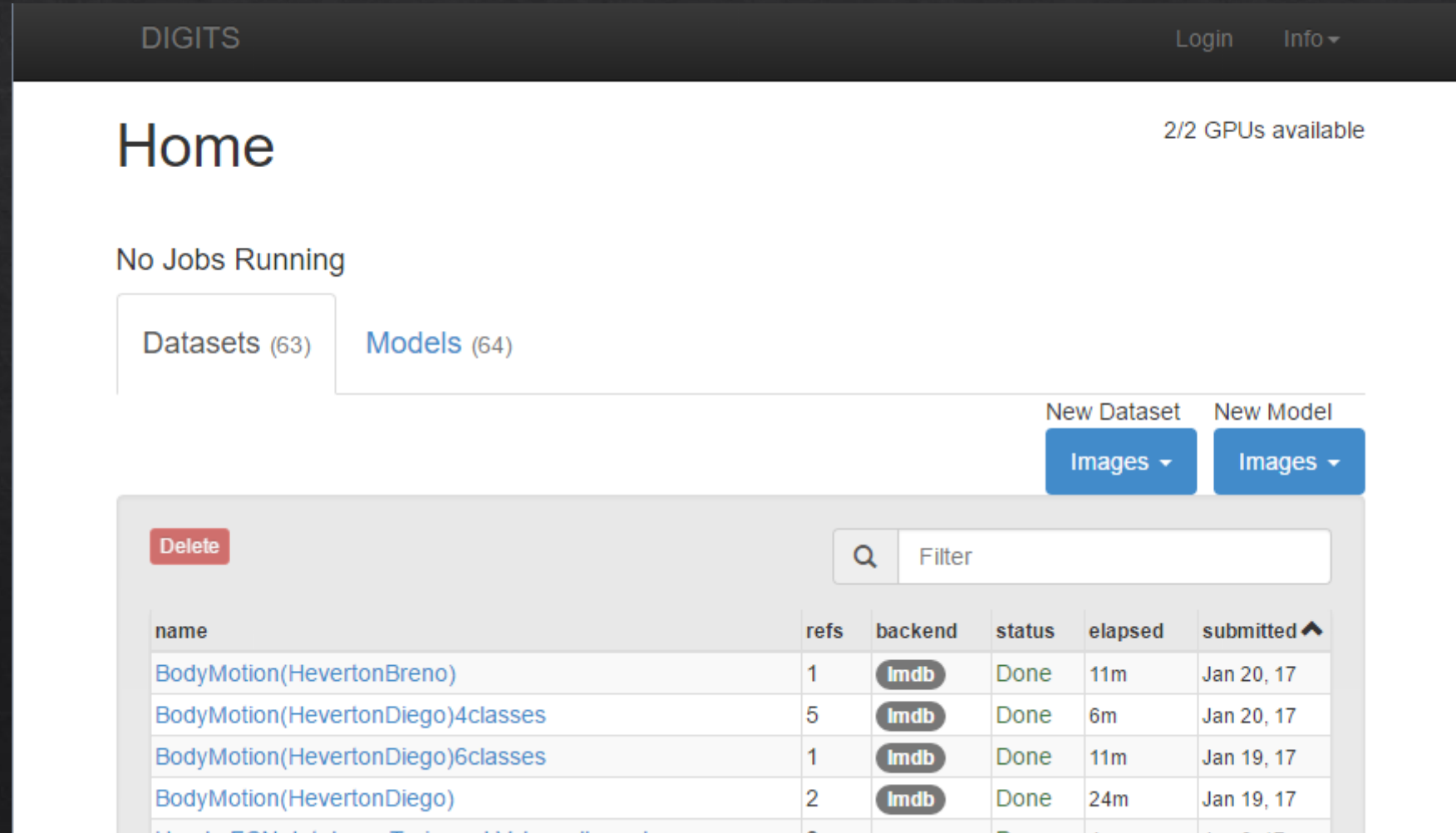
MNIST

- ◆ Dataset de dígitos
- ◆ Contém 60 000 imagens
- ◆ Cada imagem contém um único dígito centralizado
- ◆ As imagens possuem tamanho 28 x 28 pixels



Utilizando o Digits - MNIST

- ◇ Inicializar o Digits com o comando:
 - ◇ `./digits-server`
- ◇ Abrir o browser de navegação no endereço localhost
- ◇ Criar uma nova base de dados para classificação de imagens.



The screenshot shows the Digits web interface. At the top, there's a header with 'DIGITS' on the left and 'Login' and 'Info' on the right. Below the header, the main content area has 'Home' on the left and '2/2 GPUs available' on the right. A status message 'No Jobs Running' is displayed. Below this, there are two buttons: 'Datasets (63)' and 'Models (64)'. To the right, there are two blue buttons: 'New Dataset Images' and 'New Model Images'. At the bottom, there's a table with a 'Delete' button and a search filter. The table has columns: name, refs, backend, status, elapsed, and submitted. The table contains several rows of dataset information.

name	refs	backend	status	elapsed	submitted
BodyMotion(HevertonBreno)	1	lmbd	Done	11m	Jan 20, 17
BodyMotion(HevertonDiego)4classes	5	lmbd	Done	6m	Jan 20, 17
BodyMotion(HevertonDiego)6classes	1	lmbd	Done	11m	Jan 19, 17
BodyMotion(HevertonDiego)	2	lmbd	Done	24m	Jan 19, 17
Implement FCN database Train and Val smaller val	2		Done	1m	Jan 19, 17

Utilizando o Digits - MNIST

◇ Dados das Imagens:

- ◇ Grayscale
- ◇ 28 x 28
- ◇ Squash

◇ Caminho para imagens

DIGITS New Dataset bruno (Logout) Info About

Use Image Folder Use Text Files

Image Type ?
Grayscale

Image size (Width x Height) ?
28 x 28

Resize Transformation ?
Squash

See example

Training Images ?
/home/mnist/train

Minimum samples per class ?
2

Maximum samples per class ?

% for validation ?
25

% for testing ?
0

Separate validation images folder

Separate test images folder

citations.csv citations.ris Exibir todos

Treinamento

- ◆ Escolha do Dataset
- ◆ Informações de treinamento:
- ◆ Época
- ◆ Snapshot
- ◆ Batch Size
- ◆ Learning Rate

The screenshot shows a web interface for training a new image classification model. The page title is "New Image Classification Model". The interface is divided into three main sections: "Select Dataset", "Solver Options", and "Data Transformations".

Select Dataset

- Dropdown menu: Digital_Hands_NoCrop_512x256_10, MnistTest, Digital_Hands_NoCrop_512x256_50, Digital_Hands_NoCrop_512x256_25, Digital_Hands_NoCrop_512x256_50, Digital_Hands_NoCrop_512x256_50
- MnistTest**
- Done May 13, 10:13:36 PM
- Image Size**: 28x28
- Image Type**: GRAYSCALE
- DB backend**: lmbd
- Create DB (train)**: 45000 images

Solver Options

- Training epochs**: 30
- Snapshot interval (in epochs)**: 1.0
- Validation interval (in epochs)**: 1.0
- Random seed**: [none]
- Batch size**: multiples allowed

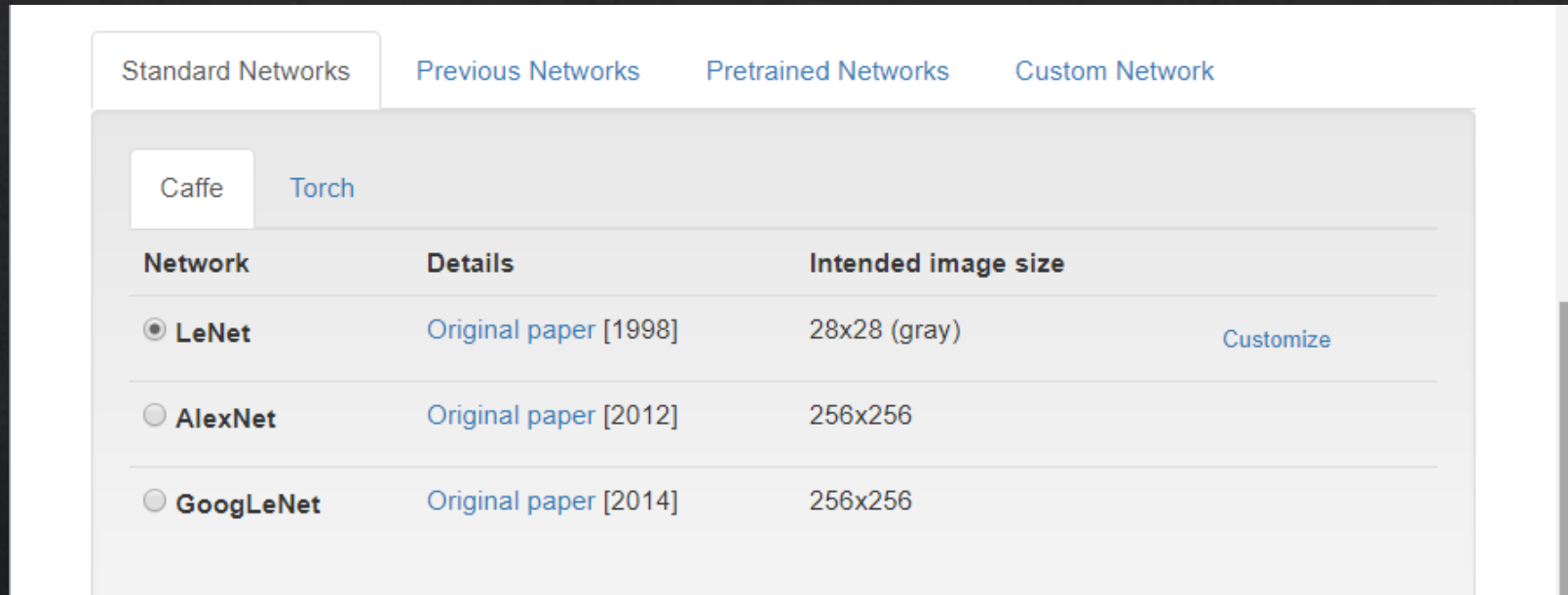
Data Transformations

- Subtract Mean**: Image
- Crop Size**: none

The interface also includes a navigation bar with "DIGITS" and "New Model" tabs, and user information "bruno (Logout) Info About". At the bottom, there is a taskbar with "citations.csv" and "citations.ris" files, and a button "Exibir todos".

Escolhendo a rede

◇ Rede padrão LeNet

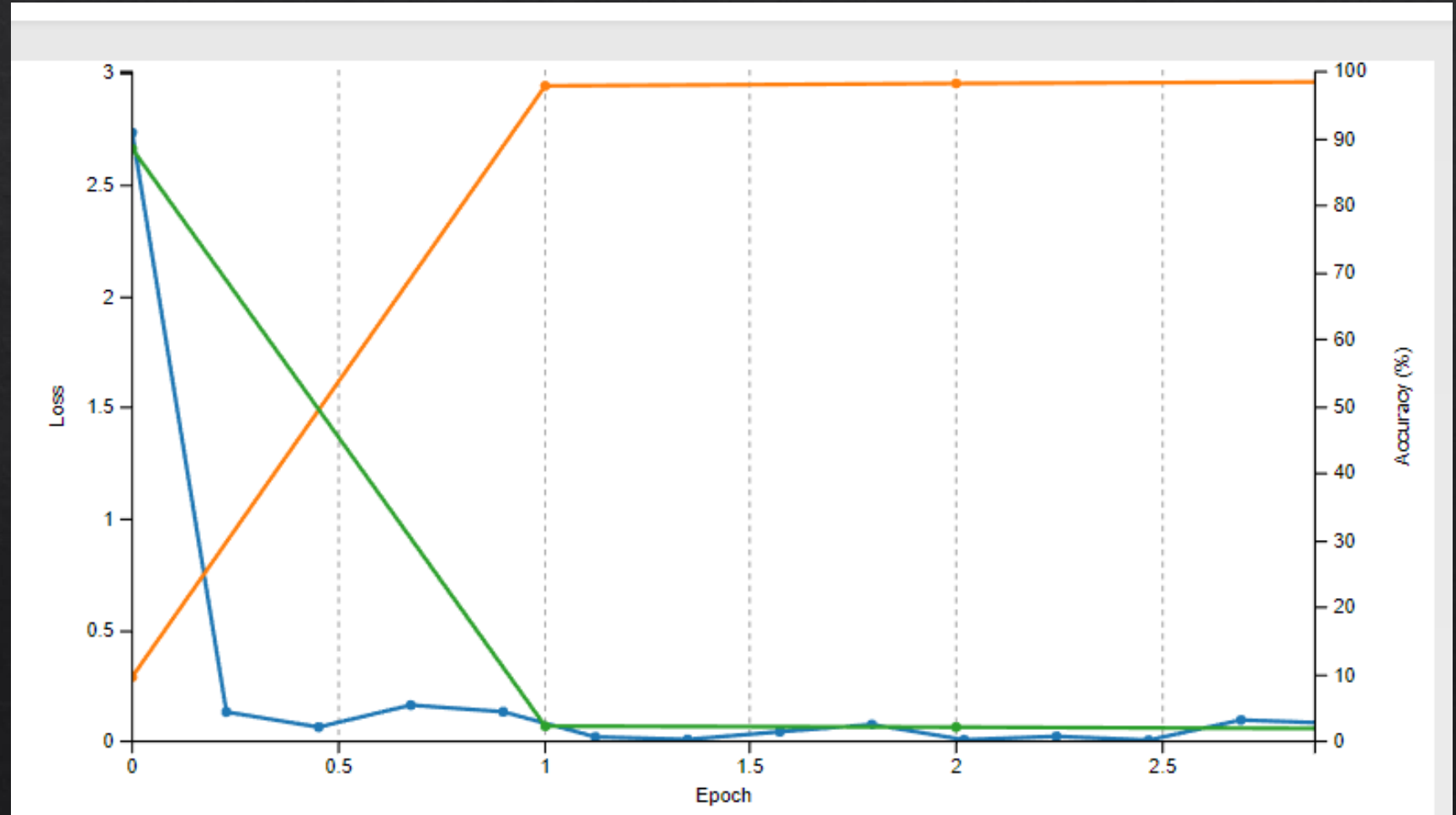


The screenshot shows a web interface for selecting a neural network. The interface has four tabs: 'Standard Networks', 'Previous Networks', 'Pretrained Networks', and 'Custom Network'. The 'Standard Networks' tab is active. Below the tabs, there are two sub-tabs: 'Caffe' and 'Torch'. The 'Caffe' sub-tab is active. Below the sub-tabs, there is a table with three columns: 'Network', 'Details', and 'Intended image size'. The table lists three networks: 'LeNet', 'AlexNet', and 'GoogLeNet'. The 'LeNet' row is selected, indicated by a radio button. The 'LeNet' row also has a 'Customize' link. The 'AlexNet' and 'GoogLeNet' rows are not selected.

Network	Details	Intended image size
<input checked="" type="radio"/> LeNet	Original paper [1998]	28x28 (gray) Customize
<input type="radio"/> AlexNet	Original paper [2012]	256x256
<input type="radio"/> GoogLeNet	Original paper [2014]	256x256

Treinamento

◇ 98% de acerto !



Testando a rede

- ❖ Testando uma única imagem ou conjunto de imagens.
- ❖ Para uma única imagem:
 1. Selecionar a época de treinamento
 2. Selecionar caminho da imagem ou fazer upload de uma imagem.

The screenshot shows the DIGITS web interface for testing a trained model. The page title is 'Image Classification Model'. Under the 'Trained Models' section, a dropdown menu is set to 'Epoch #3'. There are two buttons: 'Download Model' (blue) and 'Make Pretrained Model' (green). The interface is split into two columns. The left column, 'Test a single image', has an 'Image Path' input field, an 'Upload image' section with a 'Browse...' button, and a 'Classify One' button. A checkbox for 'Show visualizations and statistics' is present. The right column, 'Test a list of images', has an 'Upload Image List' section with a 'Browse...' button and a note: 'Accepts a list of filenames or urls (you can use your val.txt file)'. Below that is an 'Image folder (optional)' input field with a note: 'Relative paths in the text file will be prepended with this value before reading'. Further down is a 'Number of images use from the file' section with an input field set to 'All' and a note: 'Leave blank to use all'. At the bottom of this column are 'Classify Many' and 'Top N Predictions per Category' buttons, and an input field for 'Number of images to show per category' set to '9'.

Visualizar resultados

- ◆ Melhores 5 Classes
 - ◆ 99 % de confiança
- ◆ 431.080 parâmetros aprendidos

3

Predictions

3	99.99%
2	0.0%
8	0.0%
7	0.0%
0	0.0%

9

Predictions

9	99.75%
4	0.19%
8	0.05%
7	0.01%
3	0.0%

Classificação em uma CNN

- ◇ CNNs de Classificação
 - ◇ Distribuição de probabilidades



Predictions

Egyptian cat	47.22%
tabby, tabby cat	25.49%
tiger cat	23.37%
lynx, catamount	2.98%
Persian cat	0.17%

Classificação de imagem (VOC dataset) em uma rede Alexnet

Múltiplas classes

◇ Duas classes em uma mesma imagem

◇ 50 % gato / 50% cachorro ?



Predictions

tiger cat	38.86%
Pembroke, Pembroke Welsh corgi	29.31%
dhole, Cuon alpinus	11.82%
dingo, warrigal, warragal, Canis dingo	9.78%
tabby, tabby cat	1.89%

A rede não é capaz de dizer a localização dos objetos na imagem



Predictions

beagle	19.04%
italian greyhound	7.67%
whippet	7.62%
Walker hound, Walker foxhound	5.63%
basset, basset hound	5.15%

Segmentação de Imagens



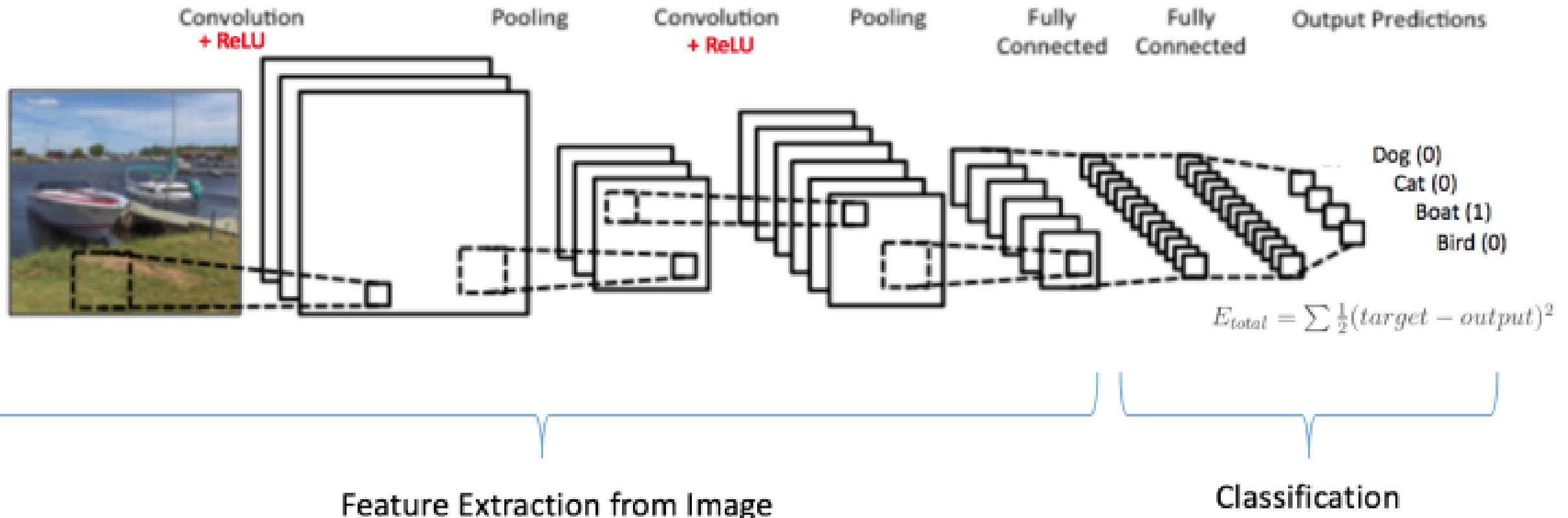
Segmentação de Imagens

- ◆ Imagens são divididas em blocos.
 - ◆ Podemos separar em nível de pixel.
- ◆ Cada bloco recebe sua própria distribuição de probabilidades



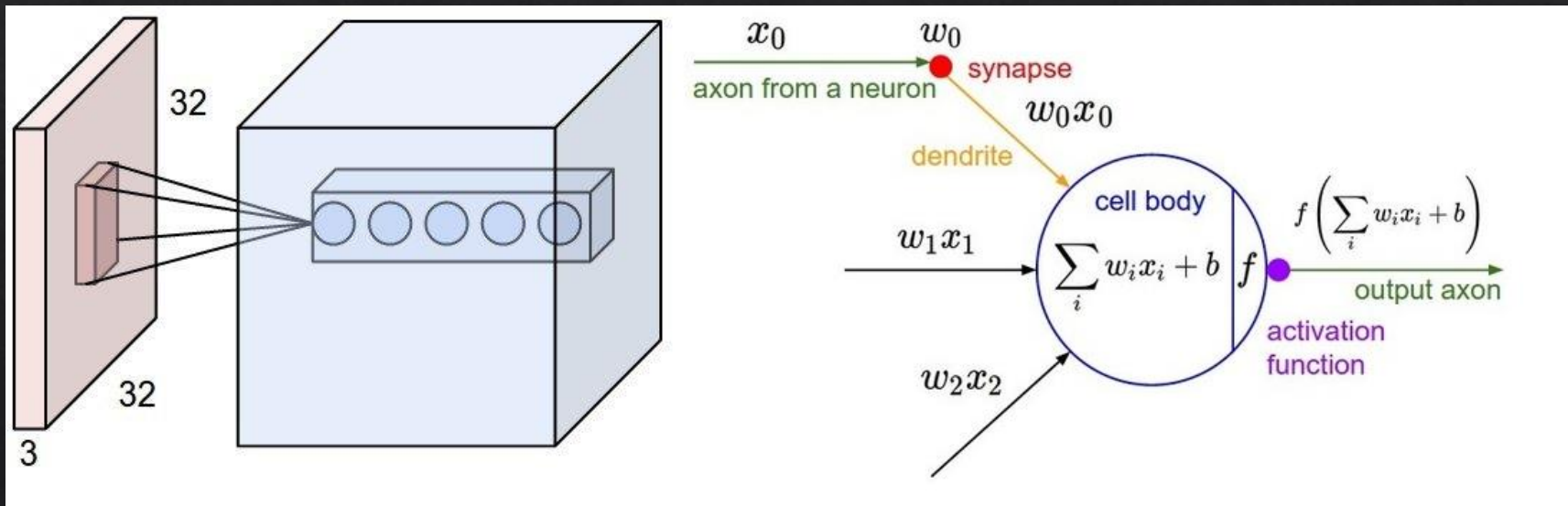
Convertendo uma CNN para Segmentação

- ◆ Uma distribuição de probabilidades por imagem → Uma distribuição de probabilidades por pixel.
- ◆ Converter uma CNN para uma *Fully Convolutional Network* (FCN)



Convertendo uma CNN para FCN

- ◇ Camada totalmente conectada:
 - ◇ Todos os neurônios de saída estão conectados a todos os neurônios de entrada.
- ◇ Camada de convolução:
 - ◇ Neurônios da camada de convolução estão conectados a um volume (campo receptivo) na entrada dos dados.



Convertendo uma CNN para FCN

- ◇ Substituir a camada totalmente conectada por uma camada de convolução equivalente.
- ◇ Tamanho do Filtro = Tamanho da entrada da camada totalmente conectada
- ◇ Número de filtros = Número de neurônios na camada totalmente conectada

```
layer {
  name: "fc6"
  type: "InnerProduct"
  bottom: "pool5"
  top: "fc6"
  inner_product_param {
    num_output: 4096
  }
}
```

```
layer {
  name: "conv6"
  type: "Convolution"
  bottom: "pool5"
  top: "conv6"
  convolution_param {
    num_output: 4096
    kernel_size: 6
  }
}
```

Camada deconvolução

- ◇ O tamanho da segmentação da FCN depende do tamanho da imagem de entrada. Segmentação por pixel.
- ◇ As camadas de convolução comprimem informações para gerar features de alto nível.
- ◇ Desejamos interpolar essas features para realizar uma classificação para cada pixel da imagem.
- ◇ Operação inversa a convolução.
 - ◇ Convolução diminui a imagem
 - ◇ Deconvolução aumenta a imagem

Fator de deconvolução

- ◆ Para cada camada de convolução e pooling, devemos calcular o fator de escala.
- ◆ Fator de escala = $1/\text{stride}$
- ◆ A escala final será a escala acumulativa de todas as camadas de convolução e pooling.

```
layer {  
  name: "conv1"  
  type: "Convolution"  
  bottom: "data_preprocessed"  
  top: "conv1"  
  convolution_param {  
    num_output: 96  
    kernel_size: 11  
    stride: 4 }  
}
```

Fator de deconvolução

◇ AlexNet -> FCN8

Layer	Stride	Pad	Kernel size	Scaling Factor (1/S)	Cumulative Scaling Factor
conv1	4	100	11	1/4	1/4
pool1	2	0	3	1/2	1/8
conv2	1	2	5	1	1/8
pool2	2	0	3	1/2	1/16
conv3	1	1	3	1	1/16
conv4	1	1	3	1	1/16
conv5	1	1	3	1	1/16
pool5	2	0	3	1/2	1/32
conv6	1	0	6	1	1/32
conv7	1	0	1	1	1/32
upscore	32	0	63	32*	1

Camada de final de deconvolução

```
layer {  
  name: "upscore"  
  type: "Deconvolution"  
  bottom: "score"  
  top: "upscore"  
  convolution_param {  
    num_output: 12 # number of classes  
    kernel_size: 63  
    stride: 32 }  
}
```

Model Store

[Update Model List](#)


Name	Contributor	Affiliate	Note	Data sets	License
Model Store Staging Server					
LeNet	NVIDIA		DIGITS example	MNIST	Apache License 2.0
AlexNet	NVIDIA		Accuracy: top-1=0.584679, top-5=0.905106	ILSVRC2012	Unrestricted
GoogleNet	NVIDIA		Accuracy: top-1=0.720625, top-5=0.969375	ILSVRC2012	Unrestricted
InceptionV1	NVIDIA		Accuracy: top-1=0.712899, top-5=0.969375	ILSVRC2012	Creative Commons License
InceptionV3	NVIDIA		Accuracy: top-1=0.84375, top-5=0.984375	ILSVRC2012	Creative Commons License
VGG16 FP32	NVIDIA		Accuracy: top-1=0.75, top-5=0.95	ILSVRC2012	Creative Commons Attribution
autoencoder	NVIDIA			MNIST	Unrestricted
siamese	NVIDIA			MNIST	Unrestricted
semantic-segmentation	NVIDIA			PASCAL VOC	Unrestricted
FCN-Alexnet	NVIDIA		FCN-Alexnet pre-trained on ILSVRC2012	ILSVRC2012	BSD-2
finetune	NVIDIA			MNIST	Unrestricted
Import AlexNet	NVIDIA			KITTI	
fcn8-semantic-segmentation	NVIDIA			PASCAL VOC	BSD-2
Epithelium segmentation	NVIDIA			Epithelium (Case Western ...	BSD-2
Sunnybrook left ventricle segmentation	NVIDIA			Sunnybrook left ventricle MR	BSD-2
ResNet-50	NVIDIA			ILSVRC2012	BSD-2

Dataset SYNTHIA

Cenas urbanas geradas artificialmente

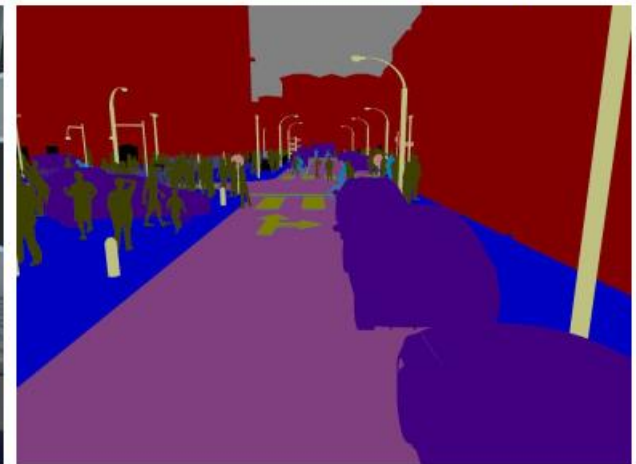
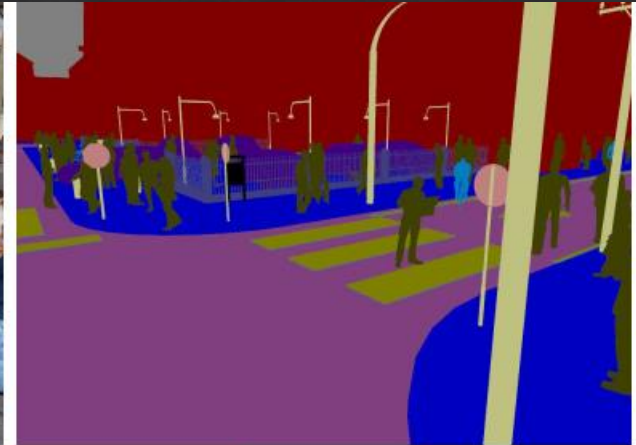
Classes como :

Construções

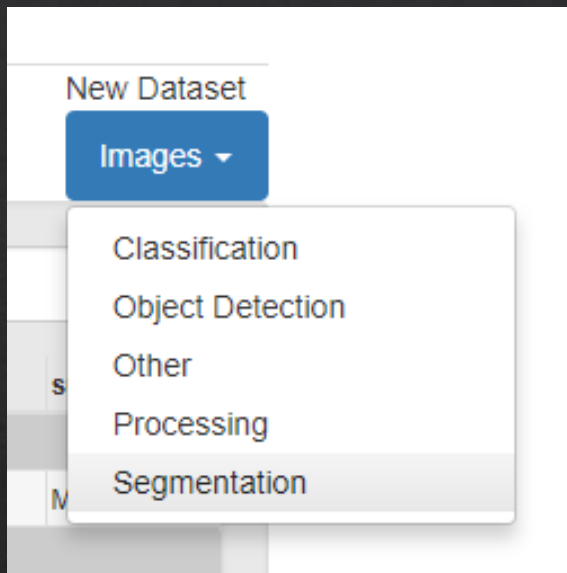
Pedestres

Carros

Sinais de trânsito



Referência: Ros, German, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez; [“The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes.”](#) *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016*: 3234-3243.

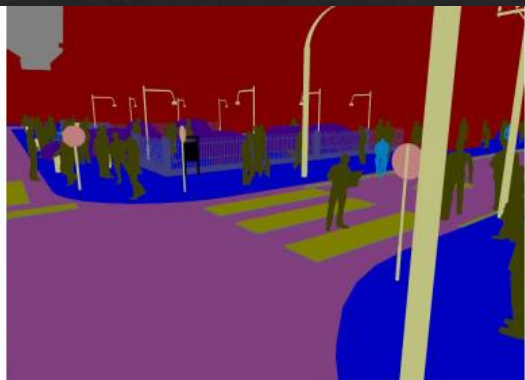


Feature



/feature/img1.jpg

Label



/label/img1.jpg

Feature image folder ?

/home/badmarques/synthia/feature/

Label image folder ?

/home/badmarques/synthia/label/

% for validation ?

25

Separate validation images

Class labels (optional) ?

file

Color map specification ?

From label image

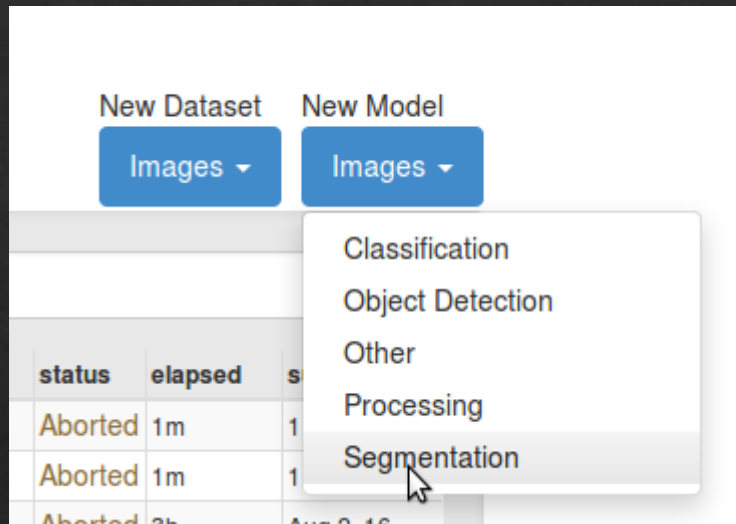
Color map file ?

file

Channel conversion ?

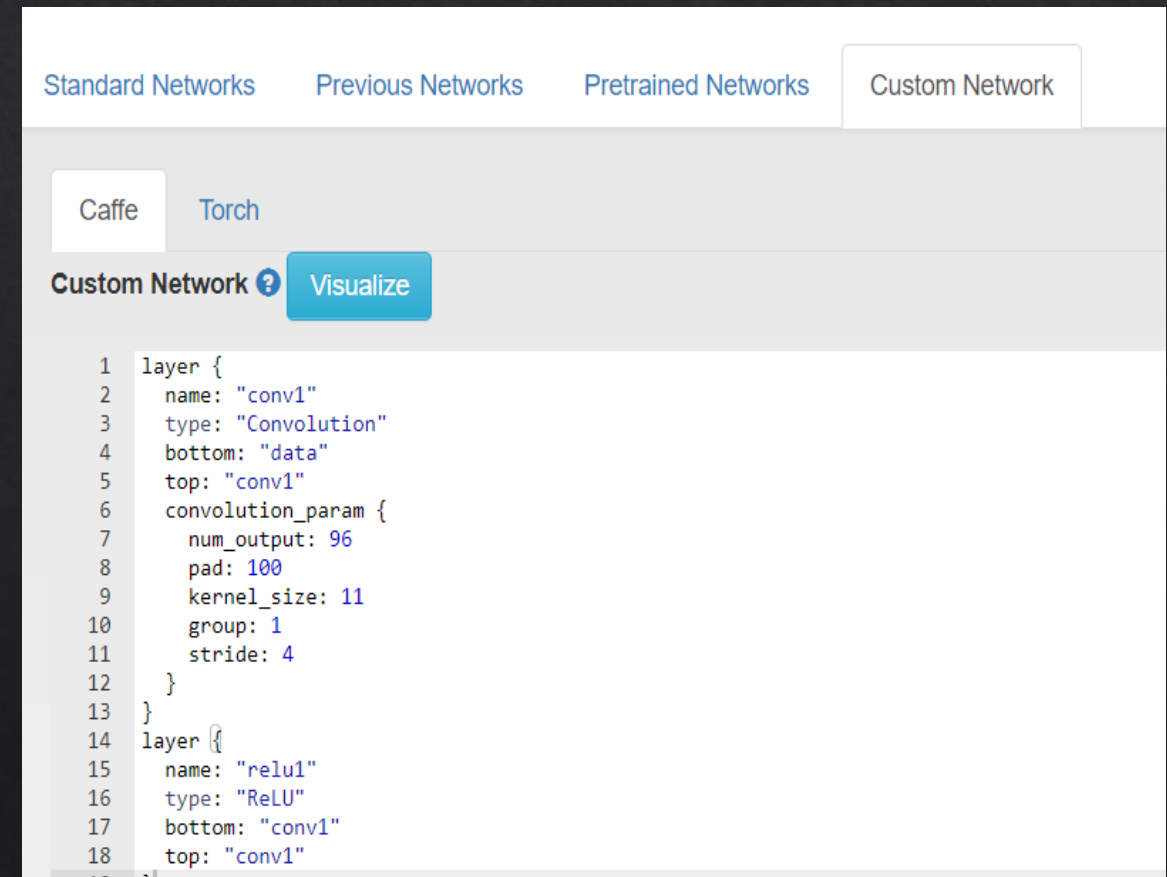
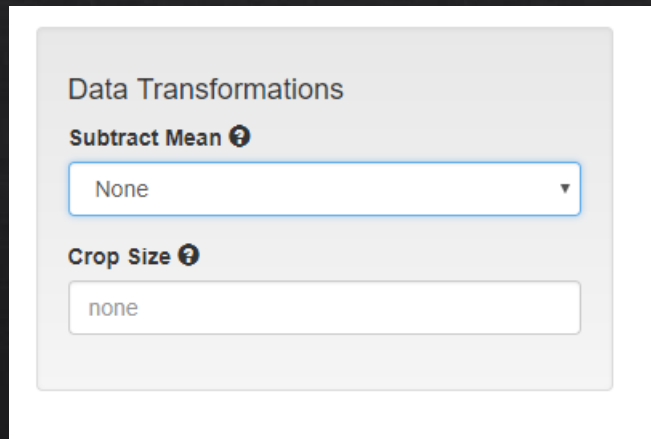
None

Treinando uma rede de segmentação



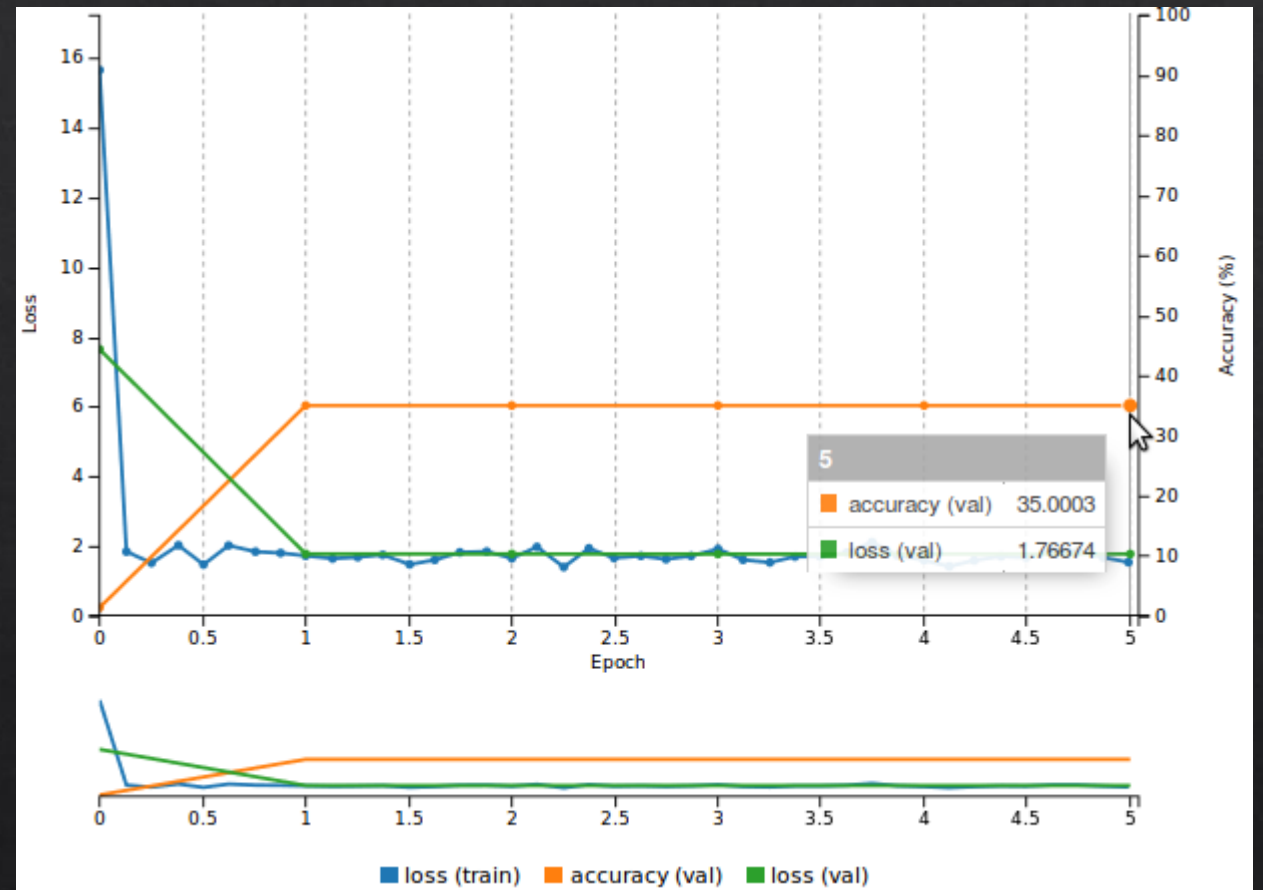
- ❖ Selecione Custom Network e adicione a **definição** da rede FCN no campo abaixo:

- ❖ O campo Subtract Mean deve ser **none**.



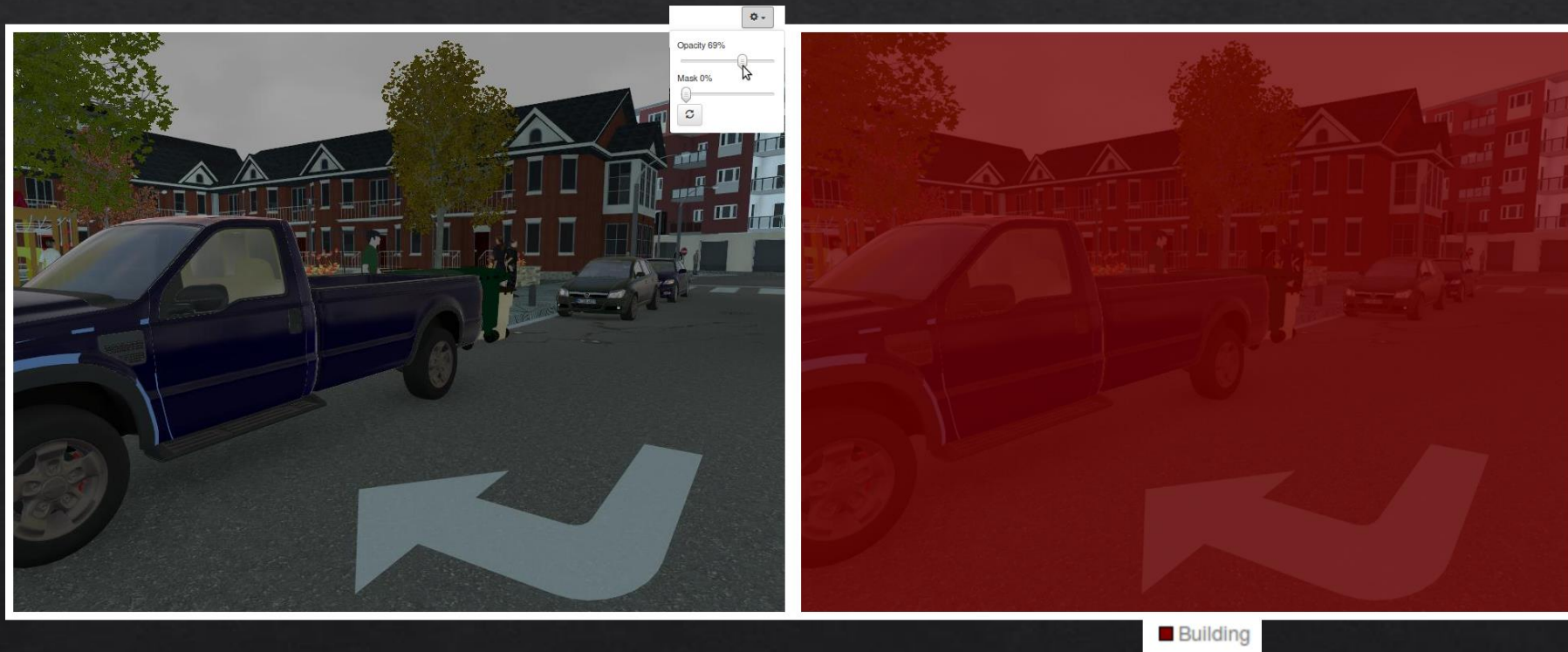
Treinamento

- ◇ Underfitting
 - ◇ A rede convergiu.
 - ◇ Acurácia de ~35%
 - ◇ Erro de treinamento aproximadamente igual ao de validação



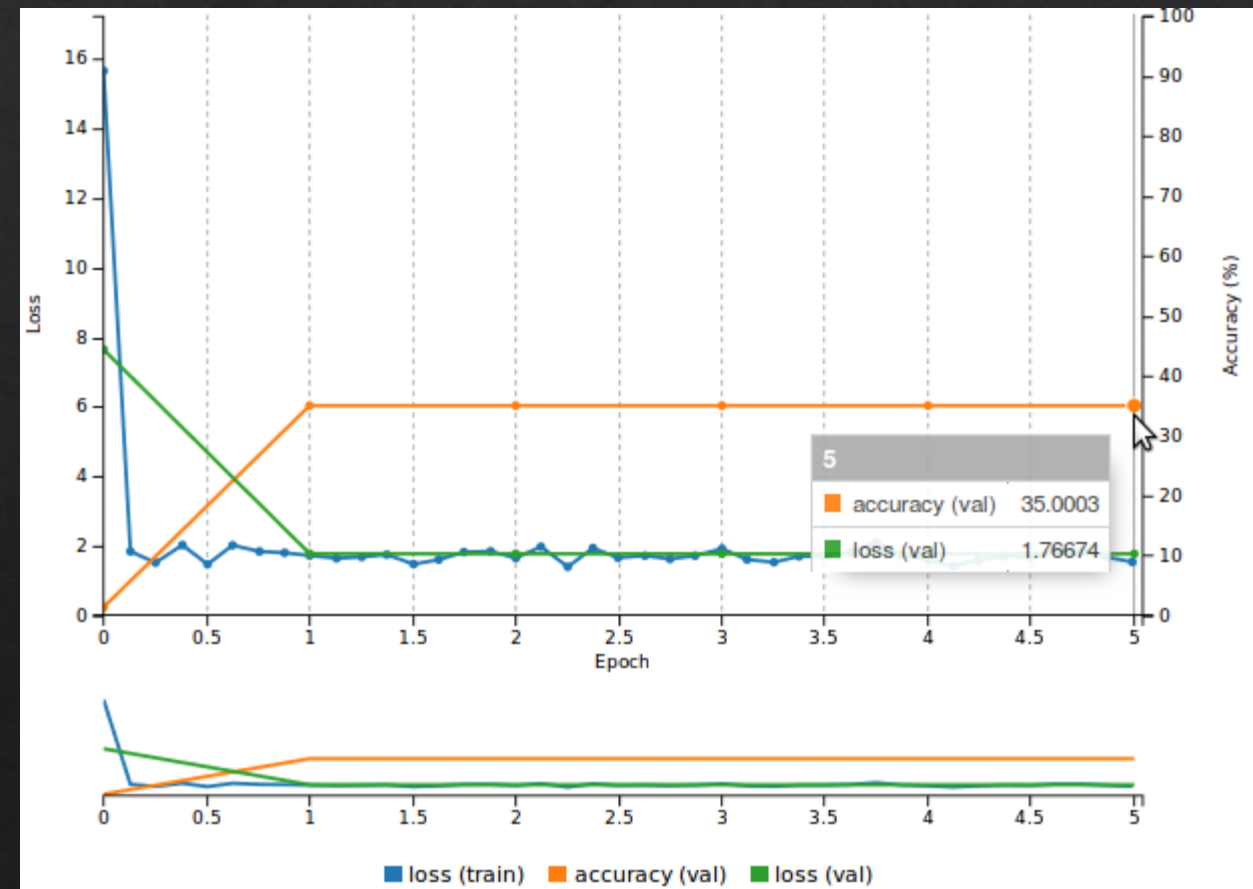
Teste

- ◆ Dataset desbalanceado. Construção é a classe com maior ocorrência no dataset.
- ◆ Rede classifica tudo como Construção



Soluções comuns

- ◇ Aumentar o tempo de treinamento
 - ◇ Solução pouco provável. A rede encontrou um mínimo local e não consegue sair dele.
- ◇ Aumentar a taxa de aprendizagem
 - ◇ Ajuda a rede a sair de um mínimo local, porém aumenta a chance de divergência.
- ◇ Aumentar o tamanho do modelo.
 - ◇ Modelos mais com mais profundos podem ser mais adequados a solução.



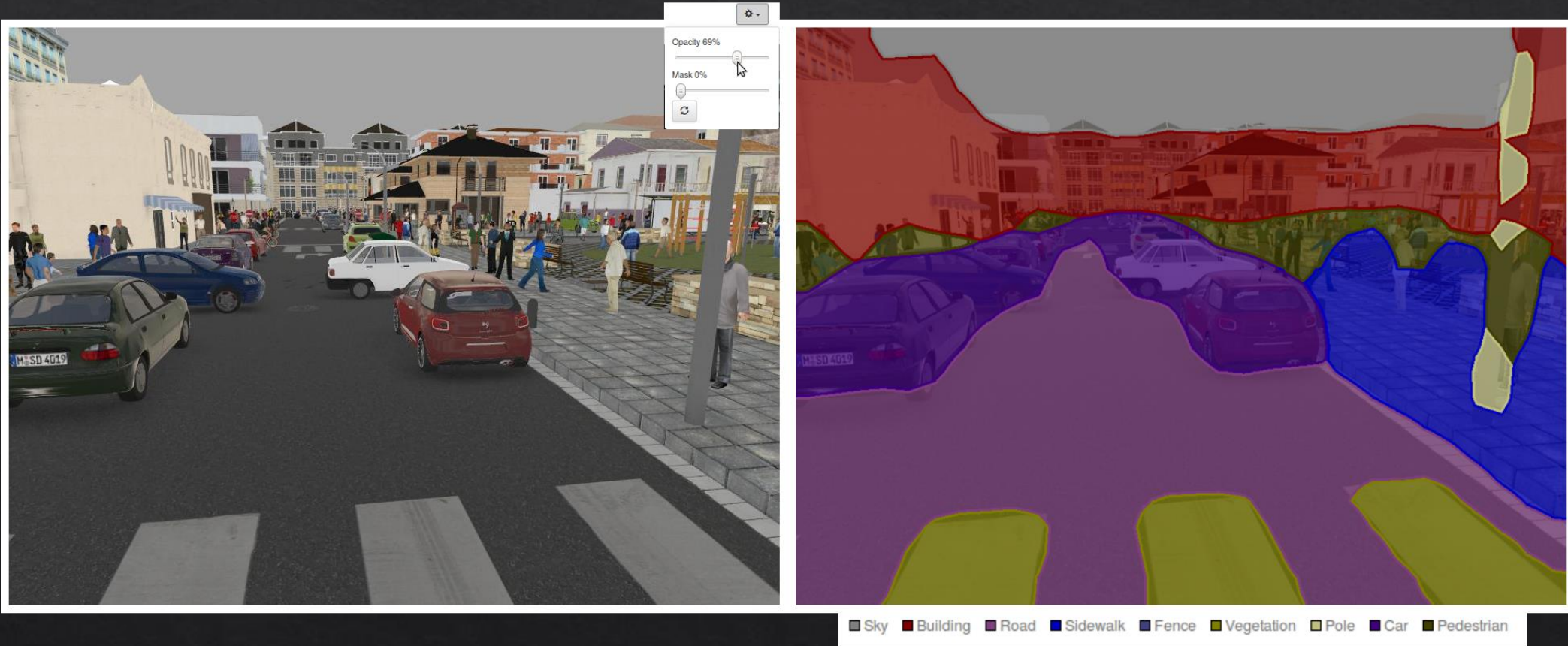
Transfer Learning

- ◆ Reutilizar o conhecimento aprendido pela rede de outros datasets.
 - ◆ Features de baixo nível (detectar bordas, linhas, formas, texturas) podem ser utilizados em todos os problemas.
- ◆ Inicializar os pesos de cada camada com os pesos aprendidos em treinamento passado.
- ◆ As redes devem possuir camadas iguais.
- ◆ Camadas com mesmo nome e parâmetros terão os pesos copiados do treinamento passado
- ◆ Camadas com nomes ou parâmetros diferentes serão inicializadas com pesos aleatórios.

Pretrained model(s) ?

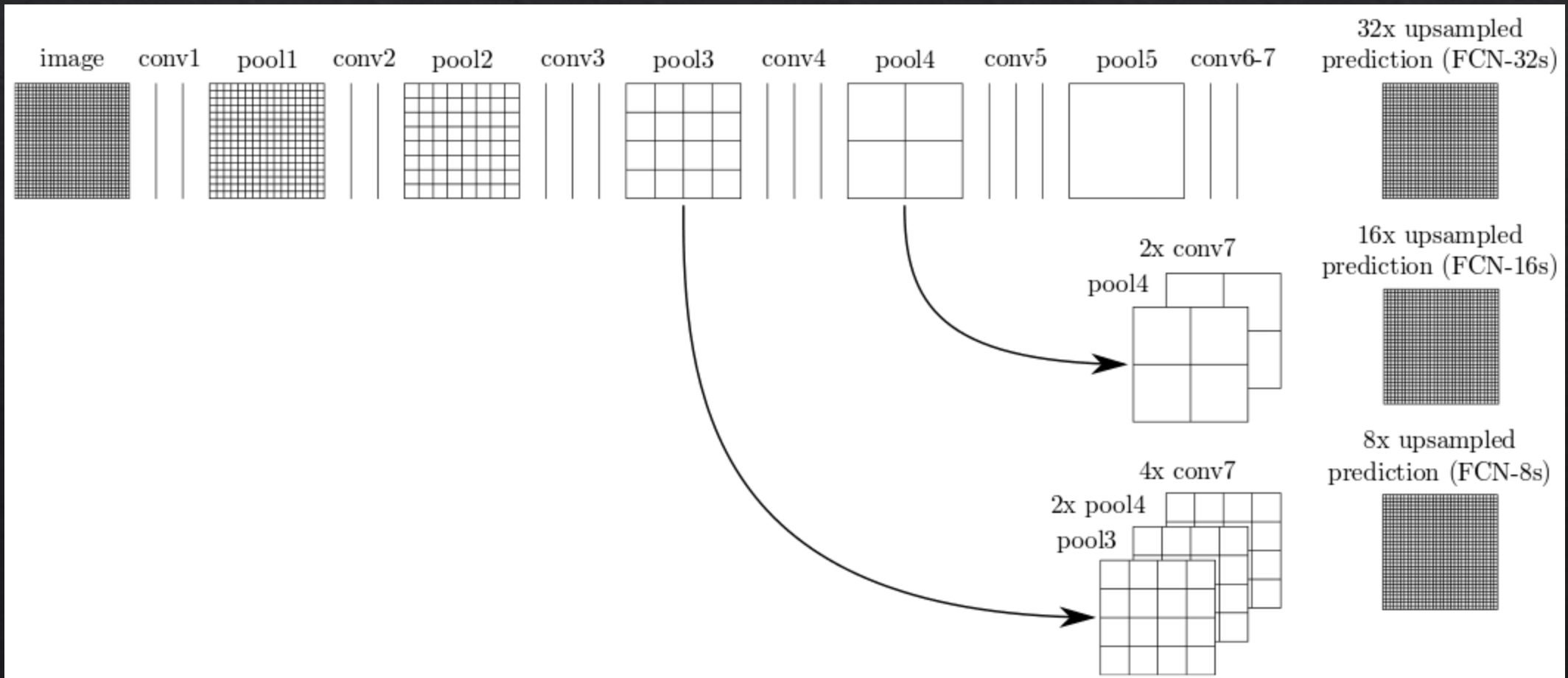
/home/bmarques/fcn/weights.prototxt

Transfer Learning no Digits



Segmentação Fina

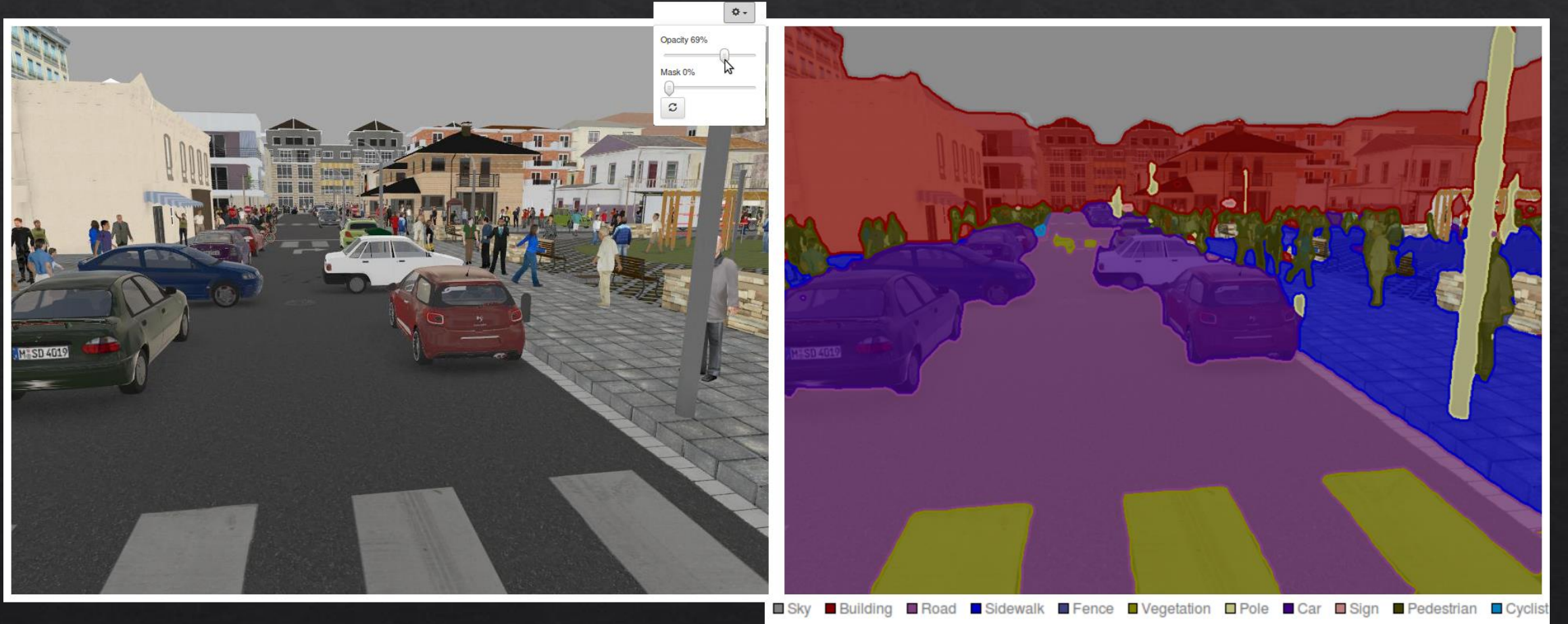
◇ “Skip connections”

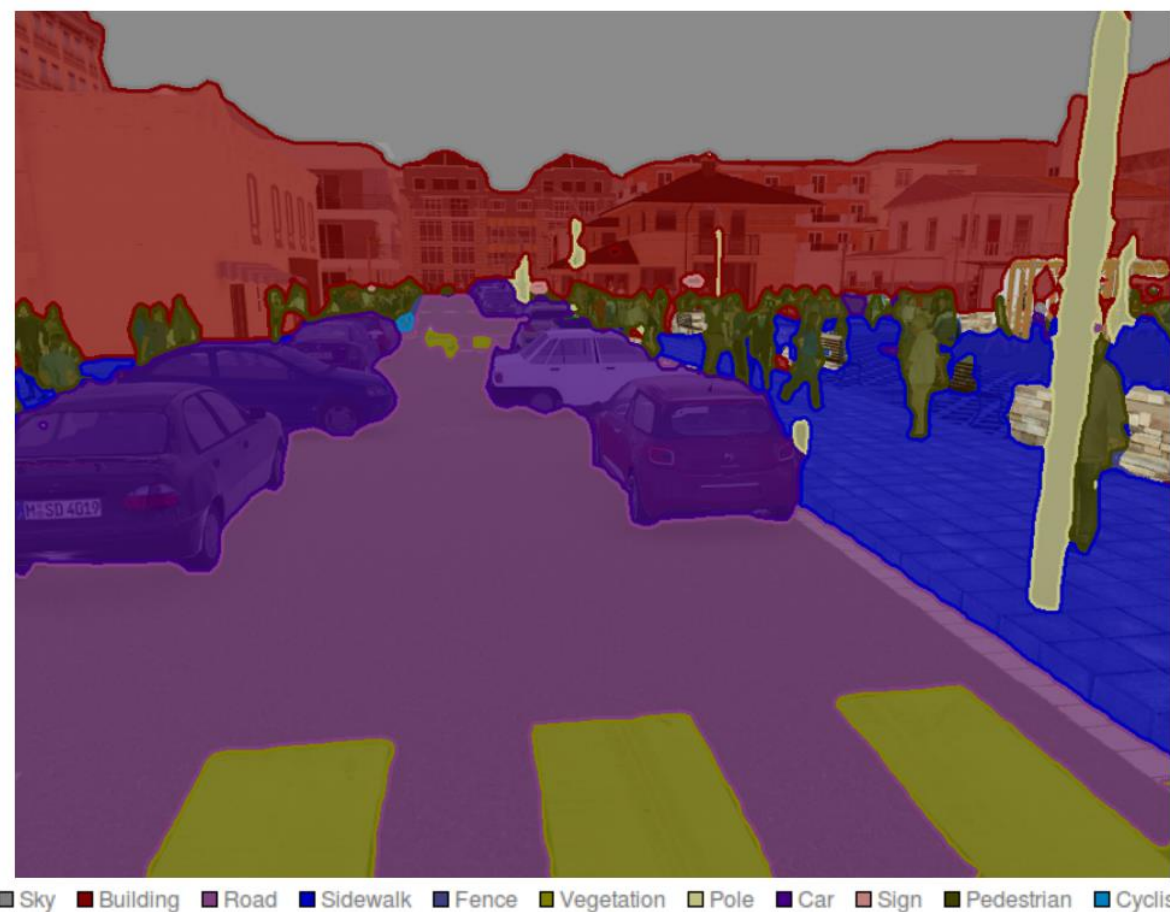
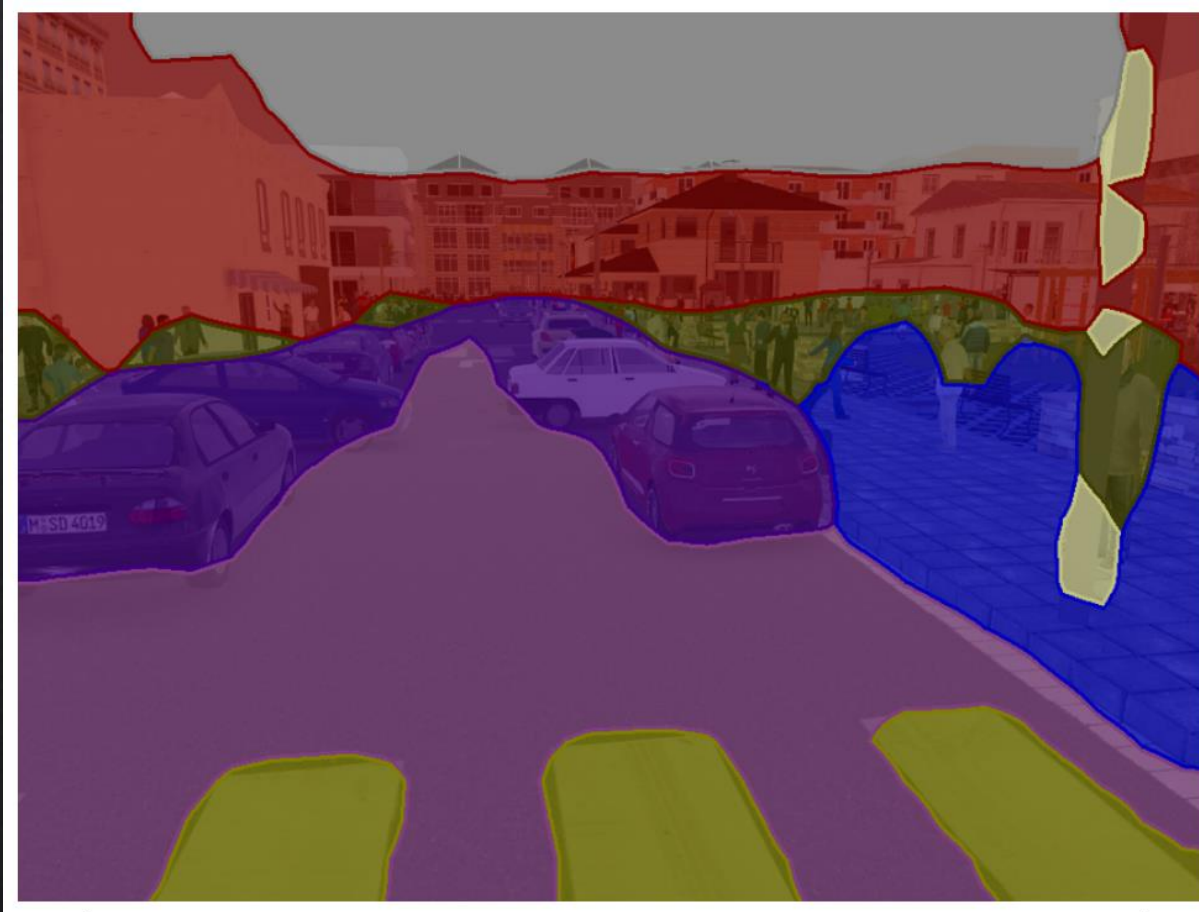


Model Store

- ◇ A rede pré treinada FCN-8s está disponível no Digits Model Store.
- ◇ O treinamento da rede é capaz de alcançar uma eficácia de 95% no dataset SYNTHIA

FCN-8s





■ Sky ■ Building ■ Road ■ Sidewalk ■ Fence ■ Vegetation ■ Pole ■ Car ■ Sign ■ Pedestrian ■ Cyclist

Referências

- ◇ Long, Jonathan, Evan Shelhamer, and Trevor Darrell. “[Fully convolutional networks for semantic segmentation.](#)” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015*: 3431-3440.
- ◇ Ros, German, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez; “[The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes.](#)” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016*: 3234-3243.
- ◇ DIGITS: Deep Learning GPU Training System - <https://devblogs.nvidia.com/parallelforall/digits-deep-learning-gpu-training-system/>
- ◇ Using DIGITS to train a Semantic Segmentation neural network - <https://github.com/NVIDIA/DIGITS/tree/digits-5.0/examples/semantic-segmentation>
- ◇ Image Segmentation Using DIGITS 5 - <https://devblogs.nvidia.com/parallelforall/image-segmentation-using-digits-5/>



Perguntas ?

BRUNO AUGUSTO DORTA MARQUES

brunodortamarques@gmail.com

brunodorta@id.uff.br