DEEP LEARNING FOR COMPUTER VISION
MUBARAK SHAH
CENTER FOR RESEARCH IN COMPUTER VISION (CRCV)

shah@crcv.ucf.edu
http://crcv.ucf.edu/
CONTENTS

PART-I: Deep Learning: A Short Overview

PART II: Computer Vision Employing Deep Learning
PART-I: Deep Learning: A Short Overview

CAP6412
Advanced Computer Vision

Yogesh S Rawat
What will we cover?

• Convolutional Neural Network
  • Their widespread use in deep learning

• Case study – AlexNet

• Network Training

• Recurrent Neural Networks
  • Working with sequential data
Convolutional Neural Network (CNN)

- A class of Neural Networks
  - Takes image as input
  - Make predictions about the input image

Source: https://adeshpande3.github.io
History

- The LeNet architecture (1990s)

Gradient-based learning applied to document recognition
First Strong Results

• AlexNet 2012
  • Winner of ImageNet Large-Scale Visual Recognition Challenge (ILSVRC 2012)
  • Error rate – 15.4% (the next best entry was at 26.2%)

Imagenet classification with deep convolutional neural networks
Today: CNNs are everywhere

Classification

Source: http://karpathy.github.io
Today: CNNs are everywhere

Object detection

Faster R-CNN: Ren, He, Girshick, Sun 2015

Semantic Segmentation

Today: CNNs are everywhere

Image captioning

Style transfer

"Show and tell: A neural image caption generator.“

A Neural Algorithm of Artistic Style
L. Gatys et al. 2015.
CNN – Not just images

• Natural Language Processing (NLP)
  • Text classification
  • Word to vector

• Audio Research
  • Speech recognition
  • Can be represented as spectrograms

• Converting data to a matrix (2-D) format
  • 1D convolution – Audio, EEG, etc.
  • 3D convolution - Videos

https://adeshpande3.github.io
CNN – Is it perfect?

Nguyen A, Yosinski J, Clune J. Deep Neural Networks are Easily Fooled: (CVPR '15), IEEE, 2015
Convolutional Neural Network
Neural Network vs CNN

• Neural Network
  • Fully connected layers

• Image as input in neural network
  • Size of feature vector = HxWxC
  • For 256x256 RGB image
    • 196,608 dimensions

• CNN - Special type of neural network
  • Operate with volume of data
  • Weight sharing in form of kernels

Source: http://cs231n.github.io
Convolution

• Core building block of a CNN
  • Spatial structure of image is preserved

A filter/kernel is convolved with the image
Convolution

• Convolution at one spatial location

32x32x3 image

3x3x3 filter

Result of convolution
Convolution

- Convolution over whole image

32x32x3 image

32x32x3 filter

Convolve over all spatial locations

Activation map (feature map)
Convolution

• Multiple filters

32x32x3 image

Convolve over all spatial locations

2 3x3x3 filter

Activation maps (feature maps)

32x32x3 image

32

32

32

30

30

30
Convolution

• One convolution layer
  • 6 3x3x3 kernels

32x32x3 image → Convolution layer → Activation maps

3 32 32 → 6 30 30
Convolutional Network

• Convolution network is a sequence of these layers

![Diagram of convolutional network layers](image)
Convolutional Network

- Convolution network is a sequence of these layers

![Diagram of a convolutional network with dimensions and filters specified.](image-url)
Convolution Operation

• Convolution of two functions \( f \) and \( g \)

\[
(f * g)(t) = \int_{-\infty}^{+\infty} f(\tau)g(t - \tau)d\tau
\]

In CNN we use 2D convolutions
Demo

Input image

filter

output
Demo

Input image

filter

output
Pooling

- Introduce translation invariance
- Makes the representations smaller
- Operates over each activation map independently

Source: http://cs231n.github.io
Activation Functions

- Introduces Non-Linearity

\[
\sigma(x) = \frac{1}{1 + e^{-x}}
\]

Sigmoid

tanh

\[
\tanh(x)
\]

ReLU

\[
\max(0, x)
\]
Convolution - Intuition

Source: https://cs.nyu.edu/~fergus/tutorials/deep_learning_cvpr12/
Visualizing Convolution

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

Source: http://cs231n.github.io
AlexNet: Network Size

- Input 227x227x3
- 5 convolution layers
- 3 dense layers
- Output 1000-D vector
AlexNet: Network Size

- Input: 227x227x3 images
- First layer (CONV1): 96 11x11 filters applied at stride 4
- What is the output volume size? \( \frac{(227-11)}{4} + 1 = 55 \)
- What is the number of parameters? \( 11 \times 11 \times 3 \times 96 = 35K \)
AlexNet: Network Size

- After CONV1: 55x55x96
- Second layer (POOL1): 3x3 filters applied at stride 2
- What is the output volume size? \((55-3)/2+1 = 27\)
- What is the number of parameters in this layer? 0
AlexNet: Network Size

- After POOL1: 27x27x96
- Third layer (NORM1): Normalization
- What is the output volume size? 27x27x96
AlexNet: Network Size

1. [227x227x3] INPUT
2. [55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
3. [27x27x96] MAX POOL1: 3x3 filters at stride 2
4. [27x27x96] NORM1: Normalization layer
5. [27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
6. [13x13x256] MAX POOL2: 3x3 filters at stride 2
7. [13x13x256] NORM2: Normalization layer
8. [13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
9. [13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
10. [13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
11. [6x6x256] MAX POOL3: 3x3 filters at stride 2
12. [4096] FC6: 4096 neurons
13. [4096] FC7: 4096 neurons
14. [1000] FC8: 1000 neurons (class scores)
Network Training
Convolutional Neural Network (CNN)

Source: https://adeshpande3.github.io
Loss Function

• Way to define how good the network is performing
  • In terms of prediction

• Network training (Optimization)
  • Find the best network parameters to minimize the loss

\[
L(W) = \frac{1}{N} \sum_{i=1}^{N} L_i(f(x_i, W), y_i)
\]
Loss Functions

• Cross entropy

\[-\frac{1}{N} \sum_{i=1}^{N} (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i))\]

• Kullback–Leibler divergence (KL divergence)

\[KL(P||Q) = \sum_{i} P(i) \log\left(\frac{P(i)}{Q(i)}\right)\]
Recurrent Neural Network
Recurrent Neural Network

- Process sequences
- Feedback from previous time step

An unrolled recurrent neural network.

http://colah.github.io/posts/2015-08-Understanding-LSTMs/
Recurrent Neural Network

one to one

Vanilla Neural Networks

http://karpathy.github.io
Recurrent Neural Network

one to one

one to many

Image captioning, image to words

http://karpathy.github.io
Recurrent Neural Network

one to one  one to many  many to one

Sentiment Classification
sequence of words -> sentiment

http://karpathy.github.io
Recurrent Neural Network

http://karpathy.github.io
Recurrent Neural Network

Video classification on frame level

http://karpathy.github.io
Long-Short Term Memory (LSTM)

- Three main components
  - Forget gate layer

http://colah.github.io/posts/2015-08-Understanding-LSTMs/
Long-Short Term Memory (LSTM)

• Three main components
  • Forget gate layer
  • Input gate layer

http://colah.github.io/posts/2015-08-Understanding-LSTMs/
Long-Short Term Memory (LSTM)

- Three main components
  - Forget gate layer
  - Input gate layer
  - Output gate layer

http://colah.github.io/posts/2015-08-Understanding-LSTMs/
CNN-LSTM

http://jeffdonahue.com/lrcn/

Vinyals et. al. Show and Tell, 2015
Jef et. al. Long-term Recurrent Convolutional Networks 2015
CNN-LSTM

http://jeffdonahue.com/lrcn/

Vinyals et. al. Show and Tell, 2015
Jef et. al. Long-term Recurrent Convolutional Networks 2015
CNN-LSTM

http://jeffdonahue.com/lrcn/

Vinyals et. al. Show and Tell, 2015
Jef et. al. Long-term Recurrent Convolutional Networks 2015
Summary

- CNN
  - Convolution
  - Pooling
  - RELU
- Case study – AlexNet
- Network Training
  - Loss Function
- Recurrent Neural Networks
  - Working with sequential data
PART-I: Deep Learning: A Short Overview

CAP6412
Advanced Computer Vision

Yogesh S Rawat
Questions?
Contents

• PART-I: Deep Learning: A Short Overview

• PART II: Computer Vision Employing Deep Learning
PART II: Computer Vision
Employing Deep Learning
DEEP LEARNING
A SHORT OVERVIEW
## Classical Computer Vision vs Deep Learning

<table>
<thead>
<tr>
<th>Classical Computer Vision</th>
<th>Deep Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Encode Expert Knowledge into constraints</td>
<td>2. Real Learning</td>
</tr>
<tr>
<td>3. Convert constraints into Objective Function</td>
<td>3. Requires Labeled/Annotated Data</td>
</tr>
<tr>
<td>4. Optimize Objective Function</td>
<td>4. Requires Massively Parallel Computations (GPUs)</td>
</tr>
<tr>
<td></td>
<td>5. Use simple Stochastic Gradient Descend</td>
</tr>
</tbody>
</table>
MAIN THEMES

GAN: Generative-Adversarial Network
Reinforcement Learning
Transfer Learning/Domain Adaptation
Multi-modal Analysis
End-to-End Learning
Bayesian Deep Learning
Deep Convolutional Neural Network (DCNN)

- A class of Neural Networks
  - Takes image as input
  - Make predictions about the input image

Source: https://adeshpande3.github.io
CONTENTS

Semantic Segmentation

Facial Attributes Detection

Human Re-Identification

Human Action Localization

Video Fill In The Blank

Reading The Mind
CONTENTS

Sematic Segmentation

Facial Attributes Detection

Human Re-Identification

Target Detection in WAMI

Anomaly Detection
CONTENTS

- Semantic Segmentation
- Facial Attributes Detection
- Human Re-Identification
- Target Detection in WAMI
- Anomaly Detection
- Human Action Localization
- Video Fill In The Blank
- Reading The Mind

Single Blank:
He ___ up the steps of the stand and away. (Runs)
Semi Supervised Semantic Segmentation Using Generative Adversarial Network

Nasim Souly, Concetto Spampinato and Mubarak Shah

ICCV 2017

SEMANTIC SEGMENTATION (SCENE LABELLING)
Assigning a semantic label to each pixel of an image.
Motivation

- Lack of enough annotated data
- Plentiful unlabeled data
- Use generative model to improve classifiers
SEMI SUPERVISED LEARNING (SSL)

Halfway between supervised and unsupervised learning

Data points lying on the same feature manifold are more expected to be classified into the same class

Leverage the unlabeled data to find this structure.

Cost function for SSL

\[
\text{Loss} = \sum_{n=1}^{N_l} \text{Loss}_l(y_n, x_n) + w \sum_{n=1}^{N_u} \text{Loss}_u(x_n)
\]
**GENERATIVE ADVERSARIAL NETWORK**

Enables models to tackle unsupervised learning

The intuitive idea:
- A painter who wants to do art forgery (G), (of Picasso)
- Someone is judging paintings (D)
- Then G produces paintings in an attempt to fool D
- D starts learning more about Picasso, G has a harder time fooling D
- D gets really good in telling apart what is Picasso and what is not?
- G gets really good at forging Picasso paintings

From Kdnuggets http://www.kdnuggets.com
GAN

Constant competition between two networks:
- a generator ($G$) and
- discriminator ($D$).

$G$ starts from some noise, $z$, generate images $G(z)$.

$D$ takes images from the distribution (real) and fake (from $G$) and classifies them: $D(x)$ and $D(G(z))$.

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(D(x))] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$
Labels are not available for all training images,
  • leverage the unlabeled data by estimating a proper prior.
This prior is used by a classifier to improve.

In GAN:
  • Unlabeled data belongs to the same distribution of labeled data
  • Generated (fake) data does not.
SEMI SUPERVISED LEARNING USING GANS
SEMI SUPERVISED LEARNING USING GANS
DISCRIMINATOR (CLASSIFIER) NETWORK

VGG 16 network with deconvolution layers.
GENERATOR NETWORK

Noise: 100D from uniform distribution.

Feature maps: 768, 384, 256, 192 and 3.
OPTIMIZATION: DISCRIMINATOR

 Discriminator Loss:

- Unlabeled data
- Supervised Loss
- Fake images Loss

\[ L_D = - \mathbb{E}_{x \sim p_{data}}(x) \log(D(x)) + \gamma \mathbb{E}_{x,y \sim p(y,x)}[\text{CE}(y, P(y|x, D))] - \mathbb{E}_{z \sim p_z(z)} \log(1 - D(G(z))) \]
OPTIMIZATION: DISCRIMINATOR

Discriminator Loss:

- Unlabeled data
- Supervised Loss
- Fake (generated) images belonging to data distribution

\[
\mathcal{L}_D = -\mathbb{E}_{x \sim p_{\text{data}}(x)} \log(D(x)) + \gamma \mathbb{E}_{x,y \sim p(y,x)} [\text{CE}(y, P(y|x, D))] - \mathbb{E}_{z \sim p_z(z)} \log(1 - D(G(z)))
\]

\[
D(x) = [1 - P(y = \text{fake}|x)]
\]

\[
P(y = k|x, D) = \frac{e^{D_k(x)}}{\sqrt{K} \sum_{-D_k(x)}}
\]
**OPTIMIZATION**

Discriminator Loss: Supervised Loss + Unlabeled data Loss + Loss of fake generated belongs to data.

K classes - > K+1 classes (all classes plus fake)

Loss for Fake generated data - > min P(D(G(z)) | y in classes) ⇔ max P(D(G(z)) | y = fake class)

\[
\mathcal{L}_D = -\mathbb{E}_{x \sim p_{\text{data}}(x)} \log(D(x)) - \mathbb{E}_{z \sim p_z(z)} \log(1 - D(G(z))) + \gamma \mathbb{E}_{x, y \sim p(y, x)} [\text{CE}(y, P(y|x, D))],
\]

\[
D(x) = [1 - P(y = \text{fake}|x)]
\]

\[
P(y = k|x, D) = \frac{e^{D_k(x)}}{\sum_{k=1}^{K} e^{D_k(x)}}
\]
OPTIMIZATION: GENERATOR

Generator tries to generate samples close to real data

$$\min_G \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$
**OPTIMIZATION: GENERATOR**

Generator tries to generate samples close to real data

Generator Loss = \( \max P(D(G(z)) \mid y \text{ in classes}) \)

\[
\begin{align*}
\min_{G} \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \\
\max_{G} \mathbb{E}_{z \sim p_z(z)}[\log(D(G(z)))]
\end{align*}
\]
SEMI SUPERVISED LEARNING WITH WEAKLY LABELED DATA

Discriminator learns class confidences and produces

- Confidence maps for each class
- A label for the fake data.
SEMI SUPERVISED LEARNING WITH ADDITIONAL WEAKLY LABELED DATA USING CONDITIONAL GANS
SEMI SUPERVISED LEARNING USING GANS
SEMI SUPERVISED LEARNING WITH ADDITIONAL WEAKLY LABELED DATA USING CONDITIONAL GANS
SEMI SUPERVISED LEARNING WITH WEAKLY LABELED DATA

The Generator uses
- Noise
- Class label information

The Discriminator uses
- Generated data
- Unlabeled data
- Image-level labels
- Pixel-level labeled data
OPTIMIZATION IN WEAKLY SUPERVISED

Using conditional GAN

In Discriminator the unlabeled part of loss is changed

$$\min_G \max_D V(D, G) = \mathbb{E}_{x, l \sim p_{data}(x, l)}[\log(D(x, l))] + \mathbb{E}_{z \sim p_z(z, l), l \sim p_l(l)}[\log(1 - D(G(z, l), l)]$$

$$\mathcal{L}_D = -\mathbb{E}_{x, l \sim p_z(x, l)} \log[p(y = \text{fake} | x)] - \mathbb{E}_{x, l \sim p_{data}(x, l)} \log[p(y \in K_i \subset 1...k | x)] +$$

$$\gamma \mathbb{E}_{x, y \sim p(y, x)}[\text{CE}(y, P(y|x, D))].$$
OPTIMIZATION IN WEAKLY SUPERVISED

Using conditional GAN

In Discriminator the unlabeled part of loss is changed

Unlabeled Loss $\rightarrow \max P(D(x) \mid y \text{ in image level ground-truth classes})$

$$\min_G \max_D V(D, G) = \mathbb{E}_{x,l \sim p_{data}(x,l)}[\log(D(x,l))] + \mathbb{E}_{z \sim p_z(z,l), l \sim p_l(l)}[\log(1 - D(G(z,l), l))]$$

$$\mathcal{L}_D = -\mathbb{E}_{x,l \sim p_z(x,l)} \log[p(y = fake|x)] - \mathbb{E}_{x,l \sim p_{data}(x,l)} \log[p(y \in K_i \subset 1...K|x)] + \gamma \mathbb{E}_{x,y \sim p(y,x)}[\text{CE}(y, P(y|x, D))],$$
EXPERIMENTAL RESULTS:

We evaluated our method on
- PASCAL VOC 2012
- SiftFlow
- StanfordBG
- CamVid datasets.

Example: For Pascal dataset, we use all training data (1400 images) for which the pixel-level label are provided

10k additional images with image-level class labels

3 metrics: pixel accuracy, per-pixel classification accuracy and average of region intersection over union
EXPERIMENTAL RESULTS:

We evaluated our method on
- PASCAL VOC 2012
- SiftFlow
- StanfordBG
- CamVid datasets.

Example: For Pascal dataset, we use all training data (1400 images) for which the pixel-level label are provided

10k additional images with image-level class labels

3 metrics: pixel accuracy, per-pixel classification accuracy and average of region intersection over union
EXPERIMENTAL RESULTS

Datasets
- PASCAL VOC 2012
- SiftFlow
- StanfordBG
- CamVid datasets.

Evaluation metrics
- Pixel accuracy
- Per-pixel classification accuracy
- Average of region intersection over union
## Quantitative Results

### StanfordBG

<table>
<thead>
<tr>
<th>method</th>
<th>pixel accuracy</th>
<th>mean accuracy</th>
<th>mean IU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard [15]</td>
<td>73.3</td>
<td>66.5</td>
<td>51.3</td>
</tr>
</tbody>
</table>

### CamVid

<table>
<thead>
<tr>
<th>method</th>
<th>pixel accuracy</th>
<th>mean accuracy</th>
<th>mean IU</th>
</tr>
</thead>
<tbody>
<tr>
<td>SegNet(Basic) [1]</td>
<td>82.2</td>
<td>62.3</td>
<td>43.6</td>
</tr>
<tr>
<td>Image</td>
<td>Fully Supervised</td>
<td>Semi Supervised</td>
<td>Ground Truth</td>
</tr>
<tr>
<td>-------</td>
<td>------------------</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td><img src="image1.jpg" alt="Image" /></td>
<td><img src="image2.jpg" alt="Fully Supervised" /></td>
<td><img src="image3.jpg" alt="Semi Supervised" /></td>
<td><img src="image4.jpg" alt="Ground Truth" /></td>
</tr>
<tr>
<td><img src="image5.jpg" alt="Image" /></td>
<td><img src="image6.jpg" alt="Fully Supervised" /></td>
<td><img src="image7.jpg" alt="Semi Supervised" /></td>
<td><img src="image8.jpg" alt="Ground Truth" /></td>
</tr>
<tr>
<td><img src="image9.jpg" alt="Image" /></td>
<td><img src="image10.jpg" alt="Fully Supervised" /></td>
<td><img src="image11.jpg" alt="Semi Supervised" /></td>
<td><img src="image12.jpg" alt="Ground Truth" /></td>
</tr>
<tr>
<td><img src="image13.jpg" alt="Image" /></td>
<td><img src="image14.jpg" alt="Fully Supervised" /></td>
<td><img src="image15.jpg" alt="Semi Supervised" /></td>
<td><img src="image16.jpg" alt="Ground Truth" /></td>
</tr>
</tbody>
</table>
QUANTITATIVE RESULTS: PASCAL VOC 2012

Using all fully labeled and unlabeled data in train set.

<table>
<thead>
<tr>
<th>method</th>
<th>pixel accuracy</th>
<th>mean accuracy</th>
<th>mean IU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully supervised</td>
<td>90.3</td>
<td>75.9</td>
<td>62.2</td>
</tr>
</tbody>
</table>

Using 30% of fully labeled data and all unlabeled data in train set.

<table>
<thead>
<tr>
<th>method</th>
<th>pixel accuracy</th>
<th>mean accuracy</th>
<th>mean IU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully supervised</td>
<td>83.15</td>
<td>53.1</td>
<td>38.9</td>
</tr>
</tbody>
</table>
## QUANTITATIVE RESULTS: PASCAL VOC 2012

Using all fully labeled and unlabeled data in train set.

<table>
<thead>
<tr>
<th>method</th>
<th>pixel accuracy</th>
<th>mean accuracy</th>
<th>mean IU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully supervised</td>
<td>90.3</td>
<td>75.9</td>
<td>62.2</td>
</tr>
</tbody>
</table>

Using 30% of fully labeled data and all unlabeled data in train set.

<table>
<thead>
<tr>
<th>method</th>
<th>pixel accuracy</th>
<th>mean accuracy</th>
<th>mean IU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully supervised</td>
<td>83.15</td>
<td>53.1</td>
<td>38.9</td>
</tr>
</tbody>
</table>
QUANTITATIVE RESULTS: PASCAL VOC 2012

Using all fully labeled and unlabeled data in train set.

<table>
<thead>
<tr>
<th>method</th>
<th>pixel accuracy</th>
<th>mean accuracy</th>
<th>mean IU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully supervised</td>
<td>90.3</td>
<td>75.9</td>
<td>62.2</td>
</tr>
</tbody>
</table>

Using 30% of fully labeled data and all unlabeled data in train set.

<table>
<thead>
<tr>
<th>method</th>
<th>pixel accuracy</th>
<th>mean accuracy</th>
<th>mean IU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully supervised</td>
<td>83.15</td>
<td>53.1</td>
<td>38.9</td>
</tr>
</tbody>
</table>
QUALITATIVE RESULTS: VOC 2012
QUALITATIVE RESULTS: VOC 2012
QUANTITATIVE RESULTS: SIFTFLOW

Using fully labeled data and 2000 unlabeled images from SUN2012

<table>
<thead>
<tr>
<th>method</th>
<th>pixel accuracy</th>
<th>mean accuracy</th>
<th>mean IU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully supervised</td>
<td>79.2</td>
<td>40.0</td>
<td>25.8</td>
</tr>
</tbody>
</table>
QUALITATIVE RESULTS: SIFTFLOW
QUALITATIVE RESULTS: SIFTFLOW
GENERATED IMAGES
GENERATED IMAGES SIFTFLOW
GENERATED IMAGES FROM CAMVID
GENERATED IMAGES

Sky-Sea

Forest

Dog

Potted Plant

Car
Semi Supervised Semantic Segmentation Using Generative Adversarial Network

Nasim Souly, Concetto Spampinato and Mubarak Shah

ICCV 2017

Contents

- Semantic Segmentation
- Facial Attributes Detection
- Human Re-Identification
- Target Detection in WAMI
- Anomaly Detection
- Human Action Localization
- Video Fill In The Blank
- Reading The Mind
Improving Facial Attribute Prediction using Semantic Segmentation

Mahdi Kalayeh, Boqing Gong and Mubarak Shah
CVPR 2017

Problem Definition

- Facial attribute prediction
## 40 Facial Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 o Clock Shadow</td>
</tr>
<tr>
<td>Arched Eyebrows</td>
</tr>
<tr>
<td>Attractive</td>
</tr>
<tr>
<td>Bags Under Eyes</td>
</tr>
<tr>
<td>Bald</td>
</tr>
<tr>
<td>Bangs</td>
</tr>
<tr>
<td>Big Lips</td>
</tr>
<tr>
<td>Big Nose</td>
</tr>
<tr>
<td>Black Hair</td>
</tr>
<tr>
<td>Blond Hair</td>
</tr>
<tr>
<td>Blurry</td>
</tr>
<tr>
<td>Brown Hair</td>
</tr>
<tr>
<td>Bushy Eyebrows</td>
</tr>
<tr>
<td>Chubby</td>
</tr>
<tr>
<td>Double Chin</td>
</tr>
<tr>
<td>Eyeglasses</td>
</tr>
<tr>
<td>Goatee</td>
</tr>
<tr>
<td>Gray Hair</td>
</tr>
<tr>
<td>Heavy Makeup</td>
</tr>
<tr>
<td>High Cheekbones</td>
</tr>
<tr>
<td>Male</td>
</tr>
<tr>
<td>Mouth Slightly Open</td>
</tr>
<tr>
<td>Mustache</td>
</tr>
<tr>
<td>Narrow Eyes</td>
</tr>
<tr>
<td>No Beard</td>
</tr>
<tr>
<td>Oval Face</td>
</tr>
<tr>
<td>Pale Skin</td>
</tr>
<tr>
<td>Pointy Nose</td>
</tr>
<tr>
<td>Receding Hairline</td>
</tr>
<tr>
<td>Rosy Cheeks</td>
</tr>
<tr>
<td>Sideburns</td>
</tr>
<tr>
<td>Smiling</td>
</tr>
<tr>
<td>Straight Hair</td>
</tr>
<tr>
<td>Wavy Hair</td>
</tr>
<tr>
<td>Wearing Earrings</td>
</tr>
<tr>
<td>Wearing Hat</td>
</tr>
<tr>
<td>Wearing Lipstick</td>
</tr>
<tr>
<td>Wearing Necklace</td>
</tr>
<tr>
<td>Wearing Necktie</td>
</tr>
<tr>
<td>Young</td>
</tr>
</tbody>
</table>
Proposed Idea

• Exploiting semantic face parsing

From left to right: background, hair, face, eyes, eyebrows, mouth and nose

Improving Facial Attribute Prediction using Semantic Segmentation (CVPR17)
Overview

• Problem Definition: Attribute Prediction
• Current Approaches: Holistic v.s. Part-based
• Deep Learning:
  • Pass the entire image to CNN, predict all attributes jointly
  • Extract parts, extract CNN features from them, aggregate features, train multiple binary SVMs
• Part-based > Holistic
Proposed Idea

• Attributes are additive to the object
• Attributes do not appear in arbitrary regions
• Spatially decomposing objects into semantic regions
• Learning attributes in per-region fashion
• Output: attribute scores and where (spatially) they are inferred from
Semantic Segmentation Network

- An encoder-decoder de/convolutional network
- 16 layers deep architecture
- Using only 2K training data, with 7 semantic labels
Semantic Segmentation Network: Results
Basic Attribute Prediction Model

- 12 layers-deep fully convolutional network
- Blocks of convolution, batch normalization and non-linearity
- Reduce spatial resolution via max-pooling
- Global average pooling instead of FC
Basic Attribute Prediction Model

- 12 layers-deep fully convolutional network
- Blocks of convolution, batch normalization, and non-linearity
- Reduce spatial resolution via max pooling
- Global average pooling instead
Semantic Segmentation-based Gating

- Max-pooling?
- Spatial aggregation? Single label vs. Multi-label
- Restrict pooling to within regions of the same semantic label
Semantic Segmentation-based Pooling

- Global Average Pooling: agnostic w.r.t spatial permutation
- Natural correspondence to the semantic regions
- Learn the correspondence
- Localization and Detection branches
SSG: Semantic Segmentation-based Gating

- Problem with Max-pooling
- Spatial aggregation? Single label v.s. Multi-Label
- Restrict pooling to within regions of the same semantic label
SSP: Semantic Segmentation-based Pooling

- Global Average Pooling is spatially agnostic
- Attributes have natural correspondence to semantic regions
- We should learn the correspondence as well
Experiments

- Dataset: CelebA Facial Attribute Prediction
- 160K training, 20K validation, 20K testing
- 40 attributes
Experiments

<table>
<thead>
<tr>
<th>Method</th>
<th>Classification Error%</th>
</tr>
</thead>
<tbody>
<tr>
<td>PANDA [22]</td>
<td>15.00</td>
</tr>
<tr>
<td>Liu et al. [15]</td>
<td>12.70</td>
</tr>
<tr>
<td>Samangouei et al. [18]</td>
<td>10.50</td>
</tr>
<tr>
<td>Zhong et al. [23]</td>
<td>10.20</td>
</tr>
<tr>
<td>Avg. Pooling</td>
<td>9.36</td>
</tr>
<tr>
<td>SSP</td>
<td>8.98</td>
</tr>
<tr>
<td>SSG</td>
<td>9.13</td>
</tr>
<tr>
<td>SSP + SSG</td>
<td>8.84</td>
</tr>
</tbody>
</table>
Experiments

<table>
<thead>
<tr>
<th>Method</th>
<th>Classification Error%</th>
</tr>
</thead>
<tbody>
<tr>
<td>PANDA [22]</td>
<td>15.00</td>
</tr>
<tr>
<td>Liu et al. [15]</td>
<td>12.70</td>
</tr>
<tr>
<td>Samangouei et al. [18]</td>
<td>10.50</td>
</tr>
<tr>
<td>Zhong et al. [23]</td>
<td>10.20</td>
</tr>
<tr>
<td>Avg. Pooling</td>
<td>9.36</td>
</tr>
<tr>
<td>SSP</td>
<td>8.98</td>
</tr>
<tr>
<td>SSG</td>
<td>9.13</td>
</tr>
<tr>
<td>SSP + SSG</td>
<td>8.84</td>
</tr>
</tbody>
</table>

Table 2. Attribute prediction performance measured by the Classification Error% on CelebA [13] original image set. Note that FaceTracer and PANDA use groundtruth landmark points to attain face parts.

<table>
<thead>
<tr>
<th>Method</th>
<th>Classification Error%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rudd et al. [17]: Separate</td>
<td>9.78</td>
</tr>
<tr>
<td>Rudd et al. [17]: MOON</td>
<td>9.06</td>
</tr>
<tr>
<td>SPPNet*</td>
<td>9.49</td>
</tr>
<tr>
<td>Avg. Pooling</td>
<td>8.84</td>
</tr>
<tr>
<td>SSP</td>
<td>8.33</td>
</tr>
<tr>
<td>SSG</td>
<td>8.38</td>
</tr>
<tr>
<td>SSP + SSG</td>
<td>8.20</td>
</tr>
</tbody>
</table>

Table 3. Attribute prediction performance measured by the Classification Error% on CelebA [15] pre-cropped image set.

<table>
<thead>
<tr>
<th>Method</th>
<th>Original (AP%)</th>
<th>Pre-cropped (AP%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPPNet*</td>
<td>N/A</td>
<td>77.69</td>
</tr>
<tr>
<td>Avg. Pooling</td>
<td>77.05</td>
<td>79.80</td>
</tr>
<tr>
<td>SSP</td>
<td>78.01</td>
<td>81.02</td>
</tr>
<tr>
<td>SSG</td>
<td>77.46</td>
<td>80.55</td>
</tr>
<tr>
<td>SSP + SSG</td>
<td>78.74</td>
<td>81.45</td>
</tr>
</tbody>
</table>

Table 4. Attribute prediction performance of our proposed variants measured by the Average Precision (AP)% on CelebA [15] original and pre-cropped image sets.
Experiments

<table>
<thead>
<tr>
<th>Method</th>
<th>Original (AP%)</th>
<th>Pre-cropped (AP%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPPNet*</td>
<td>N/A</td>
<td>77.69</td>
</tr>
<tr>
<td>Avg. Pooling</td>
<td>77.05</td>
<td>79.80</td>
</tr>
<tr>
<td>SSP</td>
<td>78.01</td>
<td>81.02</td>
</tr>
<tr>
<td>SSG</td>
<td>77.46</td>
<td>80.55</td>
</tr>
<tr>
<td>SSP + SSG</td>
<td>78.74</td>
<td>81.45</td>
</tr>
</tbody>
</table>

Table 4. Attribute prediction performance of our proposed variants measured by the Average Precision (AP)% on CelebA [15] original and pre-cropped image sets.
Learned Correspondences
Learned Correspondences
Learning the Correspondence
Visualizing the Activations: Global average pooling
Visualizing the Activation: SSP
Effect of Attributes on Semantic Face Parsing

- We used face parsing labels to improve facial attribute prediction
- What about the other way around?
  - Can attributes improve face parsing problem?

<table>
<thead>
<tr>
<th></th>
<th>Background</th>
<th>Hair</th>
<th>Face Skin</th>
<th>Eyes</th>
<th>Eyebrows</th>
<th>Mouth</th>
<th>Nose</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o attributes</td>
<td>89.25</td>
<td>47.56</td>
<td>78.65</td>
<td>46.83</td>
<td>31.22</td>
<td>62.03</td>
<td>77.40</td>
<td>61.84</td>
</tr>
<tr>
<td>w/ attributes</td>
<td>89.64</td>
<td>48.32</td>
<td>79.92</td>
<td>56.33</td>
<td>42.25</td>
<td>65.42</td>
<td>77.74</td>
<td>65.66</td>
</tr>
</tbody>
</table>

Table 2. Effect of attributes on facial parsing in IoU(%).
Qualitative Results

<table>
<thead>
<tr>
<th>Attractive</th>
<th>Black hair</th>
<th>Brown hair</th>
<th>Male</th>
<th>Mouth slightly open</th>
<th>Smiling</th>
<th>Straight hair</th>
<th>Wearing lipstick</th>
<th>Young</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.14</td>
<td>0.03</td>
<td>0.18</td>
<td>0.49</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.00</td>
<td>0.01</td>
<td>0.15</td>
<td>0.79</td>
<td>0.07</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.77</td>
<td>0.84</td>
<td>0.17</td>
<td>0.05</td>
<td>0.15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.00</td>
<td>0.09</td>
<td>0.92</td>
<td>0.13</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.84</td>
<td>0.07</td>
<td>0.06</td>
<td>0.40</td>
<td>0.20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.84</td>
<td>0.09</td>
<td>0.00</td>
<td>0.17</td>
<td>0.71</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>0.95</td>
<td>0.95</td>
<td>0.99</td>
<td>0.97</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Qualitative Results

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attractive</td>
<td>0.54</td>
</tr>
<tr>
<td>Black hair</td>
<td>0.00</td>
</tr>
<tr>
<td>Brown hair</td>
<td>0.88</td>
</tr>
<tr>
<td>Male</td>
<td>0.01</td>
</tr>
<tr>
<td>Mouth slightly open</td>
<td>0.00</td>
</tr>
<tr>
<td>Smiling</td>
<td>0.00</td>
</tr>
<tr>
<td>Straight hair</td>
<td>0.00</td>
</tr>
<tr>
<td>Wearing lipstick</td>
<td>0.95</td>
</tr>
<tr>
<td>Young</td>
<td>0.98</td>
</tr>
</tbody>
</table>
Improving Facial Attribute Prediction using Semantic Segmentation

Mahdi Kalayeh, Boqing Gong and Mubarak Shah
CVPR 2017

Contents

- Sematic Segmentation
- Facial Attributes Detection
- Human Re-Identification
- Target Detection in WAMI
- Anomaly Detection
- Human Action Localization
- Video Fill In The Blank
- Reading The Mind
Human Semantic Parsing for Person Re-identification

Mahdi M. Kalayeh, Emrah Basaran, Muhittin Gökmen, Mustafa E. Kamasak
Mubarak Shah
CVPR 2018
http://crcv.ucf.edu/people/phd_students/mahdi/papers/CVPR18.pdf
Person Re-Identification

- Retrieving the images of a person from a large gallery
- Query and gallery images are captured by different cameras
- A cross-camera data association problem
Person Re-Identification
Challenges

- Illumination conditions
- Observable human body parts
- Perceived posture of the person
- Background clutter
- Occlusion
Extracting visual cues from human body parts

- Human pose estimation ✗
  - poor performance in low resolution images,
  - Unable to capture arbitrary contours of human body parts
- Dividing image into horizontal stripes ✗
  - Easy to implement but lacking any semantic alignment
Proposed Approach

- A simple yet effective training strategy
  - Training on aggregation of multiple datasets in low resolution (492x164)
  - Finetuning on each individual dataset in high resolution (748x246)
  - Effective use of three architectures: Inception-V3, ResNet50 and ResNet152

- Human semantic parsing
  - Addressing aforementioned challenges
**SPReID:** Human **Semantic Parsing for Person Re-identification**
SPReID - Global Representation
SPReID - Human Semantic Parsing
SPReID – Foreground & Region Representations
SPReID Representation

- Concatenation of
  - Global representation
  - Foreground representation
  - Body Part representation
Experimental Results

- **Datasets**
  - **Person Re-Identification**
    - CUHK03
    - Market-1501
    - DukeMTMC-reID
  - **Semantic parsing**
    - Look into Person (LIP)
CUHK03

- **1,360 identities captured by 6 cameras.**
  - Each identity is viewed by 2 disjoint cameras.

- **13,164 person images**
  - Each identity has on average **4.8** images in each viewpoint.

- The dataset partitions:
  - Training: **1160** persons
  - Validation: **100** persons
  - Test: **100** persons
Market-1501

- **32,668** labeled bounding boxes of **1501** subjects captured by 6 cameras
  - The bounding boxes are detected DPM

- **The dataset partitions:**
  - Training: **751** persons, **12936** images
  - Query: **750** persons, **3368** images
  - Gallery: **750** persons, **19734** images
DukeMTMC-reID

- Consists of the images extracted from the DukeMTMC tracking dataset.
  - Recorded by 8 cameras
  - Hand-annotated bounding boxes

- The dataset partitions:
  - Training: **702** persons, **16,522** images
  - Query: **702** identities, **2,228** images
  - Gallery: **702** identities, **16,522** images
Look into Person (LIP)

- **50,462** pixel-wise annotated images with **19** semantic human part labels and one background label.
Training the Baselines

- **Aggregation of 10 datasets**
  - CUHK03, Market-1501, DukeMTMC-reID, 3DPeS, CUHK01, CUHK02, PRID, PSDB, Shinpuhkan, VIPeR
  - ~111,000 images of ~17,000 identities

- **Full images without semantic segmentation**
  - Training for 200K iterations using input images of size 492×164

- **Fine-tuning on evaluation datasets separately**
  - For 50K iteration on higher input resolution of 748×246
Training SPReID

- **Person Re-Id Backbone**
  - Training is done with the exact same setting as Baseline

- **Human semantic parsing**
  - Training on Look into Person (LIP) dataset
  - Different semantic regions are grouped to create 5 coarse labels
    - foreground, head, upper-body, lower-body and shoes
Human semantic parsing
CUHK03
### Experimental Results – LIP (semantic parsing)

<table>
<thead>
<tr>
<th>Method</th>
<th>Overall Accuracy</th>
<th>Mean Accuracy</th>
<th>Mean IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>SegNet</td>
<td>69.04</td>
<td>24.0</td>
<td>18.17</td>
</tr>
<tr>
<td>FCN-8n</td>
<td>76.06</td>
<td>36.75</td>
<td>28.29</td>
</tr>
<tr>
<td>DeepLabV2</td>
<td>82.66</td>
<td>51.64</td>
<td>41.64</td>
</tr>
<tr>
<td>Attention</td>
<td>83.43</td>
<td>54.39</td>
<td>42.92</td>
</tr>
<tr>
<td>DeepLabV2+SSL</td>
<td>83.16</td>
<td>52.55</td>
<td>42.44</td>
</tr>
<tr>
<td>Attention+SSL</td>
<td>84.36</td>
<td>54.94</td>
<td>44.73</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>85.07</strong></td>
<td><strong>60.54</strong></td>
<td><strong>48.16</strong></td>
</tr>
</tbody>
</table>
Experimental Results: Person Re-ID
Effect of Input Image Size

![Graphs showing re-identification rate vs. rank score for Market-1501, CUHK03, and Duke datasets with different model sizes.](Image)
- Effect of Fine Tuning for CNN Backbone
Effect of Semantic Parsing

Market-1501

CUHK03

Duke
### Results – Market-1501

#### State-of-the-art

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP (%)</th>
<th>rank-1</th>
<th>rank-5</th>
<th>rank-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPAR</td>
<td>63.4</td>
<td>81.0</td>
<td>92.0</td>
<td>94.7</td>
</tr>
<tr>
<td>JLML</td>
<td>65.5</td>
<td>85.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Basel.+LSRO</td>
<td>66.1</td>
<td>84.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SSM</td>
<td>68.8</td>
<td>82.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DaF</td>
<td>72.4</td>
<td>82.3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Chen et. Al.</td>
<td>73.1</td>
<td>88.9</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

#### Ours - without semantic parsing

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP (%)</th>
<th>rank-1</th>
<th>rank-5</th>
<th>rank-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inception-V3\textsuperscript{ft}</td>
<td>76.56</td>
<td>90.8</td>
<td>96.35</td>
<td>97.71</td>
</tr>
<tr>
<td>ResNet-152\textsuperscript{ft}</td>
<td>77.96</td>
<td>90.71</td>
<td>96.26</td>
<td>97.65</td>
</tr>
<tr>
<td>combined</td>
<td>82.87</td>
<td>93.14</td>
<td>97.27</td>
<td>98.22</td>
</tr>
<tr>
<td>SPReID\textsuperscript{ft}</td>
<td>81.34</td>
<td>92.54</td>
<td>97.15</td>
<td>98.1</td>
</tr>
<tr>
<td>+ re-ranking</td>
<td>89.99</td>
<td>94.3</td>
<td>96.35</td>
<td>97.39</td>
</tr>
<tr>
<td>ResNet-152\textsuperscript{ft} + SPReID\textsuperscript{ft}</td>
<td>83.36</td>
<td>93.68</td>
<td>97.57</td>
<td>98.4</td>
</tr>
<tr>
<td>+ re-ranking</td>
<td>90.96</td>
<td>94.63</td>
<td>96.82</td>
<td>97.65</td>
</tr>
</tbody>
</table>
### Results – CUHK03

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP (%)</th>
<th>rank-1</th>
<th>rank-5</th>
<th>rank-10</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>State-of-the-art</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSM</td>
<td>-</td>
<td>76.6</td>
<td>94.6</td>
<td>98.0</td>
</tr>
<tr>
<td>Spindle</td>
<td>-</td>
<td>88.5</td>
<td>97.8</td>
<td>98.6</td>
</tr>
<tr>
<td>DPAR</td>
<td>-</td>
<td>85.4</td>
<td>97.6</td>
<td>99.4</td>
</tr>
<tr>
<td>Chen et. Al.</td>
<td>82.8</td>
<td>86.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HydraPlus</td>
<td>-</td>
<td>91.8</td>
<td>98.4</td>
<td>99.1</td>
</tr>
<tr>
<td><strong>Ours - without semantic parsing</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inception-V3&lt;sup&gt;ft&lt;/sup&gt;</td>
<td>-</td>
<td>88.73</td>
<td>97.82</td>
<td>98.94</td>
</tr>
<tr>
<td>ResNet-152&lt;sup&gt;ft&lt;/sup&gt;</td>
<td>-</td>
<td>90.38</td>
<td>98.71</td>
<td>99.46</td>
</tr>
<tr>
<td>combined</td>
<td>-</td>
<td>92.81</td>
<td>98.9</td>
<td>99.35</td>
</tr>
<tr>
<td>SPReID&lt;sup&gt;ft&lt;/sup&gt;</td>
<td>-</td>
<td>93.89</td>
<td>98.76</td>
<td>99.51</td>
</tr>
<tr>
<td>+ re-ranking</td>
<td>-</td>
<td>96.31</td>
<td>99.25</td>
<td>99.71</td>
</tr>
<tr>
<td><strong>Ours - with semantic parsing</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ResNet-152&lt;sup&gt;ft&lt;/sup&gt; + SPReID&lt;sup&gt;ft&lt;/sup&gt;</td>
<td>-</td>
<td>94.28</td>
<td>99.04</td>
<td>99.56</td>
</tr>
<tr>
<td>+ re-ranking</td>
<td>-</td>
<td>96.22</td>
<td>99.34</td>
<td>99.7</td>
</tr>
</tbody>
</table>
## Results – Duke

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP (%)</th>
<th>rank-1</th>
<th>rank-5</th>
<th>rank-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basel. + LSRO</td>
<td>47.1</td>
<td>67.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Basel. + OIM</td>
<td>-</td>
<td>68.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Zheng et. Al.</td>
<td>49.3</td>
<td>68.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ACRN</td>
<td>52.0</td>
<td>72.6</td>
<td>84.8</td>
<td>88.9</td>
</tr>
<tr>
<td>SVDNet</td>
<td>56.8</td>
<td>76.7</td>
<td>86.4</td>
<td>89.9</td>
</tr>
<tr>
<td>Chen et. Al.</td>
<td>60.6</td>
<td>79.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Inception-V3\textsuperscript{ft}</td>
<td>63.27</td>
<td>80.48</td>
<td>88.78</td>
<td>91.65</td>
</tr>
<tr>
<td>ResNet-152\textsuperscript{ft}</td>
<td>67.02</td>
<td>83.26</td>
<td>90.93</td>
<td>92.95</td>
</tr>
<tr>
<td>combined</td>
<td>72.0</td>
<td>85.37</td>
<td>92.15</td>
<td>94.21</td>
</tr>
<tr>
<td>SPReID\textsuperscript{ft}</td>
<td>70.97</td>
<td>84.43</td>
<td>91.88</td>
<td>93.72</td>
</tr>
<tr>
<td>+ re-ranking</td>
<td>83.16</td>
<td>87.21</td>
<td>92.37</td>
<td>93.9</td>
</tr>
<tr>
<td>ResNet-152\textsuperscript{ft} + SPReID\textsuperscript{ft}</td>
<td>73.34</td>
<td>85.95</td>
<td>92.95</td>
<td>94.52</td>
</tr>
<tr>
<td>+ re-ranking</td>
<td>84.99</td>
<td>88.96</td>
<td>93.27</td>
<td>94.75</td>
</tr>
</tbody>
</table>
Conclusion

- A simple DCNN can outperform the current state-of-the-art
  - when trained properly on large number of images
- By exploiting human semantic parsing, the performance of a baseline model can be further improved
Human Semantic Parsing for Person Re-identification

Mahdi M. Kalayeh, Emrah Basaran, Muhittin G¨okmen, Mustafa E. Kamasak
Mubarak Shah
CVPR 2018

http://crcv.ucf.edu/people/phd_students/mahdi/papers/CVPR18.pdf
Contents

- Sematic Segmentation
- Facial Attributes Detection
- Human Re-Identification
- Target Detection in WAMI
- Anomaly Detection
- Human Action Localization
- Video Fill In The Blank
- Reading The Mind
Fully Convolutional Deep Neural Networks for Persistent Multi-Frame Multi-Object Detection in Wide Area Aerial Videos

Rodney LaLonde, Dong Zhang and Mubarak Shah
CVPR-2018
The Goal

Multiple Vehicle Detection

Red dots are the ground truth (x,y) annotations. (WPAFB 2009)
Ground Truth
Heat-maps

- Gaussian heat-maps have their colors inverted and $\sigma$ slightly increased for visualization purposes.
- Binary segmentation heat-maps have spots with value 1 instead of Gaussians (and 0 background).
Supervised Learning

- 2D Convolutional Layer
- ReLU Layer
- Max Pooling Layer
- Dropout Layer (50%)
- Euclidean Loss Layer
- Solver Type
  - Adam

- Deep Learning Framework
  - Caffe
  - MATLAB interface
- GPU
  - 4 NVIDIA Titan X GPUs
- Training
  - Full training on a single GPU: 2 days
Deep Purely Convolutional Neural Network

Input Patches

\( t-n \)

\( t \)

\( t+n \)

2D Conv
Filters: 32
Kernel: 15 x 15

Max Pool
Filters: 32
Kernel: 2 x 2
Stride: 2

2D Conv
Filters: 32
Kernel: 13 x 13

2D Conv
Filters: 32
Kernel: 11 x 11

2D Conv
Filters: 256
Kernel: 9 x 9

2D Conv
Filters: 256
Kernel: 7 x 7

2D Conv
Filters: 256
Kernel: 5 x 5

2D Conv
Filters: 256
Kernel: 3 x 3

2D Conv
Filters: 1
Kernel: 1 x 1

Output

Effective Receptive Field at Each Layer of the CNN

Network Output
Experiments

1. Data Creation
WPAF 2009 Dataset

- Covering an area of over **19.5 sq. km.**
- Frame rate of roughly 1.25 Hz.
- Over 315 million pixels per frame, with each pixel corresponding to roughly **1/4 meter**.
- Vehicle makes up only approximately 9 X18 pixels
  - **2.4 million** vehicles in 1,025 frames of video,
  - Over **2,000** in every frame.
- **Eight AOIs**
  - AOI-1 to AOI-4: 2,278 X2,278 size
  - AOI 34 is 4,260 X2,604. AOI 40 is 3,265X2,542. AOI 41 is 3,207X 2,892
Experiments

4. Multi-Frame Experiments: Moving Object Detection
Results AOI 34 Gaussian Heat-map
Results AOI 41
Gaussian Heat-map

Object Detection Results: AOI 41

precision
0.5 0.6 0.7 0.8 0.9 1
recall
Results AOI 41
BinSeg Heat-map
Results AOI 41
Background Sub.
Results AOI 41

Single Frame
Results AOI 40
Gaussian Heat-map
Results AOI 40
BinSeg Heat-map

Object Detection Results: AOI 40

[Graph showing precision vs. recall]
Results AOI 02

Object Detection Results: AOI 02

- Proposed Multi-Frame
- 2-frame [21]
- 3-frame [25]
- 3-frame + BF [11]
- 3-frame + N [16]
- Median BG [22]
- Median BG + GMS [18]
- Median BG + GMT [13]
- Median BG + N [23]
- Mean BG [12]
- IGMM [16]
- Inpaint [2]
Results AOI 01
Results AOI 03

Object Detection Results: AOI 03
Results AOI 04

Object Detection Results: AOI 04

precision
recall
Experiments

4. Multi-Frame Experiments: Slowing and Stopped Vehicle Detections
Stopped Vehicles

- Of the 11 state-of-the-art methods none can handle stopped vehicles.
- For evaluation all stopped vehicles were removed
  - The number of GT objects is reduced from originally 460,612 to 163,158
- Our approach can detect stopped vehicles.
  - Experiments were ran on AOI 42 with no ground-truth coordinates removed.
Results AOI 42

Object Detection Results: AOI 42

- Proposed Multi-Frame
- Prokaj & Medioni [17]
Stopped Vehicle Detection Example
Comparison of $F_1$ Scores on Eight Crop and Aligned Sections of the WPAFB 2009 Dataset

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sommer et al. [23]</td>
</tr>
<tr>
<td>Shi [22]</td>
</tr>
<tr>
<td>Liang et al. [13]</td>
</tr>
<tr>
<td>Kent et al. [12]</td>
</tr>
<tr>
<td>Aeschliman et al. [2]</td>
</tr>
<tr>
<td>Pollard &amp; Antone (3-frame + N) [16]</td>
</tr>
<tr>
<td>Saleemi &amp; Shah [21]</td>
</tr>
<tr>
<td>Xiao et al. [25]</td>
</tr>
<tr>
<td>Keck et al. [11]</td>
</tr>
<tr>
<td>Reilly et al. [18]</td>
</tr>
<tr>
<td>Pollard &amp; Antone (IGMM) [16]</td>
</tr>
<tr>
<td>Teutsch &amp; Grinberg [24]</td>
</tr>
<tr>
<td>Prokaj &amp; Medioni [17]</td>
</tr>
<tr>
<td><strong>Proposed Multi-Frame</strong></td>
</tr>
</tbody>
</table>
Conclusions

- We have proposed a novel fully convolutional neural network based method for persistent multi-frame multi-object detection in aerial videos.

- In our method, we successfully taking advantage of both appearance and motion cues and integrate them into a single detection network, trained end to end.

- We have shown comparisons with many state-of-the-art methods, and the performance improvements are relatively 5 to 16% on moving objects for multiple videos in the WPAFB 2009 dataset as measured by F1 score and nearly 50% relative improvement on persistent detections compared to [17].

- Additionally, while detections are considered true positives if they fall within 20 pixels of the ground-truth, the proposed method’s mean distance from ground truth annotations, averaged over all true positive detections, was roughly 2 pixels (0.5 m), compared to 5.5 pixels reported in [24].

- We further demonstrated that the proposed method can handle stopped vehicles well, which is often a failure case in other methods.
Fully Convolutional Deep Neural Networks for Persistent Multi-Frame Multi-Object Detection in Wide Area Aerial Videos

Rodney LaLonde, Dong Zhang and Mubarak Shah
CVPR-2018
Real-World Anomaly Detection in Surveillance videos

Waqas Sultani, Chen Chen, Mubarak Shah

Computer Vision and Pattern Recognition (CVPR), 2018
Motivation

- Over 30 Millions cameras in US
- Over 4 Billions hours of videos per week
- Manual supervision is impossible
- Automatic Analysis is highly needed
Note: we fast play or trim some videos due to their long durations.
Anomaly Detection

The goal is to timely signal an activity that deviates normal patterns.

Anomalous events:
- traffic accidents, crimes or illegal activities, etc

Anomalous events rarely occur as compared to normal activities.
Our Approach

- Learn anomalies by exploiting both normal and anomalous videos.
- Avoid annotating the anomalous clips in training videos.
- Learn anomaly through the deep multiple instance ranking framework by leveraging weakly labeled training videos:
  - A video is normal or contains anomaly somewhere, but we do not know where.
Anomaly Detection

• The goal is to timely signal an activity that deviates normal patterns.

• Anomalous events:
  • traffic accidents, crimes or illegal activities, etc

• Anomalous events rarely occur as compared to normal activities.
Our Approach

• Learn anomalies by exploiting both normal and anomalous videos.

• Avoid annotating the anomalous clips in training videos.

• Learn anomaly through the deep multiple instance ranking framework by leveraging weakly labeled training videos:
  • A video is normal or contains anomaly somewhere, but we do not know where.
Example Anomalous Videos in the Dataset
Shoplifting

Vandalism
Weakly labeled Crime Detection Framework

Ranking

\[ f(\mathcal{V}_a) > f(\mathcal{V}_n), \]

MIL Ranking

\[ \max_{i \in \mathcal{B}_a} f(\mathcal{V}_a^i) > \max_{i \in \mathcal{B}_n} f(\mathcal{V}_n^i), \]

Loss Function

\[ l(\mathcal{B}_a, \mathcal{B}_n) = \max(0, 1 - \max_{i \in \mathcal{B}_a} f(\mathcal{V}_a^i) + \max_{i \in \mathcal{B}_n} f(\mathcal{V}_n^i)) \]
Weakly labeled Crime Detection Framework

Ranking

\[ f(V_a) > f(V_n), \]

MIL Ranking

\[
\max_{i \in B_a} f(V^i_a) > \max_{i \in B_n} f(V^i_n),
\]

Loss Function

\[
l(B_a, B_n) = \max(0, 1 - \max_{i \in B_a} f(V^i_a) + \max_{i \in B_n} f(V^i_n))
\]

\[
+ \lambda_1 \sum_{i} (f(V^i_a) - f(V^{i+1}_a))^2 + \lambda_2 \sum_{i} f(V^i_a),
\]

1. Smoothness term
2. Sparsity term
Weakly labeled Crime Detection Framework
Dataset & Experimental Results
<table>
<thead>
<tr>
<th>Anomaly</th>
<th>Number of videos</th>
</tr>
</thead>
<tbody>
<tr>
<td>Burglary</td>
<td>100</td>
</tr>
<tr>
<td>Fighting</td>
<td>50</td>
</tr>
<tr>
<td>Road Accidents</td>
<td>150</td>
</tr>
<tr>
<td>Robbery</td>
<td>150</td>
</tr>
<tr>
<td>Shooting</td>
<td>50</td>
</tr>
<tr>
<td>Shoplifting</td>
<td>50</td>
</tr>
<tr>
<td>Stealing</td>
<td>100</td>
</tr>
<tr>
<td>Abuse</td>
<td>50</td>
</tr>
<tr>
<td>Arrest</td>
<td>50</td>
</tr>
<tr>
<td>Arson</td>
<td>50</td>
</tr>
<tr>
<td>Assault</td>
<td>50</td>
</tr>
<tr>
<td>Explosion</td>
<td>50</td>
</tr>
<tr>
<td>Vandalism</td>
<td>50</td>
</tr>
<tr>
<td>Normal</td>
<td>950</td>
</tr>
</tbody>
</table>

Number of videos of each category
Qualitative Results
Ground Truth

Road Accident
Quantitative Results
Comparison with state-of-art anomaly detection methods
Comparison with state-of-art anomaly detection methods
Comparison with state-of-art anomaly detection methods

Comparison with state-of-art anomaly detection methods


Comparison with state-of-art anomaly detection methods


Comparison with stat-of-art anomaly detection methods

<table>
<thead>
<tr>
<th>Method</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary Classifier</td>
<td>50.0</td>
</tr>
<tr>
<td>Hasan et al. [1]</td>
<td>50.6</td>
</tr>
<tr>
<td>Lu et al. [2]</td>
<td>65.51</td>
</tr>
<tr>
<td>Proposed w/o constraints</td>
<td>74.44</td>
</tr>
<tr>
<td><strong>Proposed w constraints</strong></td>
<td><strong>75.41</strong></td>
</tr>
</tbody>
</table>


Comparison with stat-of-art anomaly detection methods

False Alarm Rate

<table>
<thead>
<tr>
<th>Method</th>
<th>False Alarm Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hasan et al. [1]</td>
<td>27.2</td>
</tr>
<tr>
<td>Lu et al. [2]</td>
<td>3.1</td>
</tr>
<tr>
<td>Proposed</td>
<td>1.9</td>
</tr>
</tbody>
</table>

• Lower is better


Action Recognition Results
<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>C3D [1]</td>
<td>23.0</td>
</tr>
<tr>
<td>TCNN [2]</td>
<td>28.4</td>
</tr>
</tbody>
</table>


Real-World Anomaly Detection in Surveillance videos

Waqas Sultani, Chen Chen, Mubarak Shah

Computer Vision and Pattern Recognition (CVPR), 2018
Thank You!
Contents

- Semantic Segmentation
- Facial Attributes Detection
- Human Re-Identification
- Target Detection in WAMI
- Anomaly Detection
- Human Action Localization
- Video Fill In The Blank
- Reading The Mind
T-CNN for Action Detection in Videos

Rui Hou, Chen Chen, Mubarak Shah
ICCV-2017

Action Recognition

- Biking
- SalsaSpin
- LongJump
- TennisSwing
Action Detection

• Trimmed video
  • Spatio-temporal localization

• Untrimmed video
  • Temporal localization
  • Spatio-temporal localization
Action Segmentation

• Pixel-wise Spatio-Temporal localization

Golf
Overview of Tube-CNN

- Divide input video into equal length clips
- Generate tube action proposals
- Link tube proposals
- Tube of Interest Max-pooling
- Recognize and localize actions
3D ConvNet

8-frame clip

→ 3D conv 1
→ 3D conv 2
→ 3D conv 3a
→ 3D conv 3b
→ 3D conv 3a
→ 3D conv 3b
→ 3D conv 4a
→ 3D conv 4b
→ 3D conv 5a
→ 3D conv 5b
Tube of Interest Max Pooling (TOI)

- Given a video clip
- And generated Tube of Interest
- Spatially max pool to a fixed shape
- Temporally max pool to a fixed duration
Tube of Interest Max Pooling (TOI)

- Clip with a tube proposal.
- Spatial max pooling to a fixed H and W, e.g. (H, W) = (4, 4).
- Temporal max pooling to a fixed D (e.g. D = 1).
- TOI pooling.
Tube of Interest Max Pooling (TOI)

Clip with a tube proposal

Spatial cells of different sizes

Spatial max pooling to a fixed H and W, e.g. (H, W) = (4, 4)

Temporal max pooling to a fixed D (e.g. D = 1)

ToI pooling
Tube Proposal Network
Proposal Linking
## Network Details

<table>
<thead>
<tr>
<th>name</th>
<th>kernel dims $(d \times h \times w)$</th>
<th>output dims $(C \times D \times H \times W)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv1</td>
<td>$3 \times 3 \times 3$</td>
<td>$64 \times 8 \times 300 \times 400$</td>
</tr>
<tr>
<td>max-pool1</td>
<td>$1 \times 2 \times 2$</td>
<td>$64 \times 8 \times 150 \times 200$</td>
</tr>
<tr>
<td>conv2</td>
<td>$3 \times 3 \times 3$</td>
<td>$128 \times 8 \times 150 \times 200$</td>
</tr>
<tr>
<td>max-pool2</td>
<td>$2 \times 2 \times 2$</td>
<td>$128 \times 4 \times 75 \times 100$</td>
</tr>
<tr>
<td>conv3a</td>
<td>$3 \times 3 \times 3$</td>
<td>$256 \times 4 \times 75 \times 100$</td>
</tr>
<tr>
<td>conv3b</td>
<td>$3 \times 3 \times 3$</td>
<td>$256 \times 4 \times 75 \times 100$</td>
</tr>
<tr>
<td>max-pool3</td>
<td>$2 \times 2 \times 2$</td>
<td>$256 \times 2 \times 38 \times 50$</td>
</tr>
<tr>
<td>conv4a</td>
<td>$3 \times 3 \times 3$</td>
<td>$512 \times 2 \times 38 \times 50$</td>
</tr>
<tr>
<td>conv4b</td>
<td>$3 \times 3 \times 3$</td>
<td>$512 \times 2 \times 38 \times 50$</td>
</tr>
<tr>
<td>max-pool4</td>
<td>$2 \times 2 \times 2$</td>
<td>$512 \times 1 \times 19 \times 25$</td>
</tr>
<tr>
<td>conv5a</td>
<td>$3 \times 3 \times 3$</td>
<td>$512 \times 1 \times 19 \times 25$</td>
</tr>
<tr>
<td>conv5b</td>
<td>$3 \times 3 \times 3$</td>
<td>$512 \times 1 \times 19 \times 25$</td>
</tr>
<tr>
<td>toi-pool2*</td>
<td>–</td>
<td>$128 \times 8 \times 8 \times 8$</td>
</tr>
<tr>
<td>toi-pool5</td>
<td>–</td>
<td>$512 \times 1 \times 4 \times 4$</td>
</tr>
<tr>
<td>1x1 conv</td>
<td>–</td>
<td>$8192$</td>
</tr>
<tr>
<td>fc6</td>
<td>–</td>
<td>$4096$</td>
</tr>
<tr>
<td>fc7</td>
<td>–</td>
<td>$4096$</td>
</tr>
</tbody>
</table>
Training Steps

• Initialize TPN based on the pre-trained C3D model

• Use the generated proposals to initialize recognition network

• Use the weights tuned by recognition network to update TPN

• Use tuned weights and proposals from TPN for final recognition network
Implementation Details

- Clip: 400 x 300 x 8 (h x w x d)

- Learning Rate: Initialized at $10^{-3}$ and
  - decreased to $10^{-4}$ after 60k iteration.

- Weight Decay: 0.0005

- Batch Size: 4
Negative Mining

- Untrimmed Videos contain positive and negative clips
- Initialize the TPN by using only positive clips.
- Apply the trained model on the whole training video (positive and negative clips)
- Select boxes in negative clips with highest scores as hard negatives

In updating TPN procedure, we choose
  - 32 boxes, which have IoU with any ground truth greater than 0.7 as positive samples
  - Randomly pick another 16 samples as negative
  - Select 16 samples from hard negative pool as negative.
Experimental results for UCF-Sports

UCF-Sports -- Skip Pooling

<table>
<thead>
<tr>
<th>Feature From</th>
<th>Frame mAP $\alpha = 0.5$</th>
<th>Frame mAP $\alpha = 0.2$</th>
<th>Video mAP $\alpha = 0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_5$</td>
<td>74.9</td>
<td>91.6</td>
<td>77.9</td>
</tr>
<tr>
<td>$C_5 + C_1$</td>
<td>81.2</td>
<td>94.5</td>
<td>92.1</td>
</tr>
<tr>
<td>$C_5 + C_2$</td>
<td>86.7</td>
<td>95.2</td>
<td>94.8</td>
</tr>
<tr>
<td>$C_5 + C_3$</td>
<td>85.8</td>
<td>95.2</td>
<td>91.7</td>
</tr>
<tr>
<td>$C_5 + C_4$</td>
<td>77.6</td>
<td>91.3</td>
<td>81.2</td>
</tr>
</tbody>
</table>
Detection Results for UCF-Sports

Horse Riding

Golf Swing

Diving

Running

Red: Our detection
Green: Ground Truth
## UCF-Sports – Frame mAP

<table>
<thead>
<tr>
<th></th>
<th>Diving</th>
<th>Golf</th>
<th>Kick</th>
<th>Lifting</th>
<th>Riding</th>
<th>Run</th>
<th>Skate.</th>
<th>Swing</th>
<th>Swing Bench</th>
<th>Walk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gkioxari et al.</td>
<td>75.8</td>
<td>69.3</td>
<td>54.6</td>
<td>99.1</td>
<td>89.6</td>
<td>54.9</td>
<td>29.8</td>
<td>88.7</td>
<td>74.5</td>
<td>44.7</td>
</tr>
<tr>
<td>Wenzaepfel et al.</td>
<td>60.71</td>
<td>77.55</td>
<td>65.26</td>
<td>100.0</td>
<td>99.53</td>
<td>52.6</td>
<td>47.14</td>
<td>88.88</td>
<td>62.86</td>
<td>64.44</td>
</tr>
<tr>
<td>Peng et al.</td>
<td>96.12</td>
<td>80.47</td>
<td>73.78</td>
<td>99.17</td>
<td>97.56</td>
<td>82.37</td>
<td>57.43</td>
<td>83.64</td>
<td>98.54</td>
<td>75.99</td>
</tr>
<tr>
<td>Ours</td>
<td>84.38</td>
<td>90.79</td>
<td>86.48</td>
<td>99.77</td>
<td>100.00</td>
<td>83.65</td>
<td>68.72</td>
<td>65.75</td>
<td>99.62</td>
<td>87.79</td>
</tr>
</tbody>
</table>

Action Detection for J-HMDB

- BrushHair
- Claping
- Picking
- Kicking
- Golf
- Climbing Stairs

Red: Our detection
Green: Ground Truth
## Action Detection for J-HMDB

<table>
<thead>
<tr>
<th></th>
<th>f.-mAP (α=0.5)</th>
<th>v.-mAP (α=0.2)</th>
<th>v.-mAP (α=0.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gkioxari et al.</td>
<td>36.2</td>
<td>-</td>
<td>53.3</td>
</tr>
<tr>
<td>Wenzaepfel et al.</td>
<td>45.8</td>
<td>63.1</td>
<td>60.7</td>
</tr>
<tr>
<td>Peng et al.</td>
<td>58.5</td>
<td>74.3</td>
<td>73.1</td>
</tr>
<tr>
<td>Ours w/o skip</td>
<td>47.9</td>
<td>66.9</td>
<td>58.6</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>84.38</strong></td>
<td><strong>90.79</strong></td>
<td><strong>86.48</strong></td>
</tr>
</tbody>
</table>

An End-to-end 3D Convolutional Neural Network for Action Detection and Segmentation in Videos
Action Segmentation Results on J-HMDB

- Catch
- Brush hair
- Clap
- Golf
- Climb
- Pick

Red: Bounding box detection
Green: Segmentation Map
Trained on JHMDB tested on UCF Sports

Red: Bounding box detection
Green: Segmentation map
Densely Annotated Video Segmentation (DAVIS’16)

• Video Object Segmentation Dataset
• 50 Videos
• 3455 Frames
• 480P: 854X480
Video Object Segmentation on Davis’16

Swan

HorseJump-High

Libby

Car-roundabout
Comparison of Different Approaches
Comparison of Mean Jaccard Index on DAVIS’16
Qualitative Results on DAVIS’16

<table>
<thead>
<tr>
<th>Measure</th>
<th>ARP</th>
<th>FSEG</th>
<th>LMP</th>
<th>FST</th>
<th>CUT</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I )</td>
<td>Mean↑</td>
<td>76.2</td>
<td>70.7</td>
<td>70.0</td>
<td>55.8</td>
<td>55.2</td>
</tr>
<tr>
<td>Recall↑</td>
<td>91.1</td>
<td>83.5</td>
<td>85.0</td>
<td>64.9</td>
<td>57.5</td>
<td><strong>95.2</strong></td>
</tr>
<tr>
<td>Decay↓</td>
<td>7.0</td>
<td>1.5</td>
<td>1.3</td>
<td><strong>0.0</strong></td>
<td>2.2</td>
<td>2.3</td>
</tr>
<tr>
<td>( J )</td>
<td>Mean↑</td>
<td>70.6</td>
<td>65.3</td>
<td>65.9</td>
<td>51.1</td>
<td>55.2</td>
</tr>
<tr>
<td>Recall↑</td>
<td>83.5</td>
<td>73.8</td>
<td>79.2</td>
<td>51.6</td>
<td>61.0</td>
<td><strong>94.7</strong></td>
</tr>
<tr>
<td>Decay↓</td>
<td>7.9</td>
<td>1.8</td>
<td>2.5</td>
<td>2.9</td>
<td>3.4</td>
<td>4.9</td>
</tr>
<tr>
<td>( T )</td>
<td>Mean↓</td>
<td>39.3</td>
<td>32.8</td>
<td>57.2</td>
<td>36.6</td>
<td>27.7</td>
</tr>
</tbody>
</table>
# Action Detection on THUMOS’13

<table>
<thead>
<tr>
<th></th>
<th>f.-mAP (α=0.5)</th>
<th>v.-mAP (α=0.05)</th>
<th>v.-mAP (α=0.1)</th>
<th>v.-mAP (α=0.2)</th>
<th>v.-mAP (α=0.3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wenzaepfel et al.</td>
<td>35.84</td>
<td>54.3</td>
<td>51.7</td>
<td>46.8</td>
<td>37.8</td>
</tr>
<tr>
<td>Peng et al.</td>
<td>39.63</td>
<td>54.5</td>
<td>50.4</td>
<td>42.3</td>
<td>32.7</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>41.37</strong></td>
<td><strong>54.7</strong></td>
<td><strong>51.3</strong></td>
<td><strong>47.1</strong></td>
<td><strong>39.2</strong></td>
</tr>
</tbody>
</table>

Experiments for THUMOS’13

Biking

HorseRiding

FloorGymnastics

CliffDiving

Red: Our detection
Green: Ground Truth
T-CNN for Action Detection in Videos

Rui Hou, Chen Chen, Mubarak Shah

ICCV-2017

Contents

- Target Detection in WAMI
- Facial Attributes Detection
- Human Re-Identification
- Sematic Segmentation
- Anomaly Detection
- Human Action Localization
- Video Fill In The Blank
- Reading The Mind
Video Fill In the Blank using LR/RL LSTMs with Spatial-Temporal Attentions

Amir Mazaheri, Dong Zhang, and Mubarak Shah

ICCV 2017

http://crcv.ucf.edu/papers/iccv17/PID4929115.pdf
Problem Definition

• **Input:**
  • A video
  • An Incomplete Sentence

• **Output:**
  • Missing word(s)

**Single Blank:**

He ___ up the steps of the stand and away.
Block Diagram

- Incomplete Sentence
- Video
- Sentence Encoder
- Video Encoder
- Inference Module
- Missing Word
Block Diagram

- Incomplete Sentence
- Video
- Sentence Encoder
- Temporal Attention
- Spatial Attention
- Inference Module
- Missing Word
Sentence Encoder

Someone stops his ______ and kisses her on the head

Encoding by LSTM

Encoding by LSTM
Sentence Encoder

\[ u_q = \text{tanh}(W_u[u^1_L | u^1_R | u^2_R]) \]
One-hot Vector

\[ q^1_l = W^1_x[x^1, x^2, \ldots, x^{t-1}] \]
\[ q^1_r = W^1_x[x^n, x^{n-1}, \ldots, x^{t+1}] \]

Word Embedding

Left and right Encoded Fragments

\[ u^1_l = LSTM^{1}_{LR}(q^1_{l(i)}), \quad (i = 1, \ldots, (t - 1)) \]
\[ u^1_r = LSTM^{1}_{RL}(q^1_{r(i)}), \quad (i = 1, \ldots, (n - t)) \]
Left/Right Memory vectors
Built by Right/Left encoded fragments

Encoded fragment to memory vector mapping

\[ \mu_r = u^1_r W_\mu \]
\[ \mu_l = u^1_l W_\mu \]

New left and right sequences, by concatenating memory

\[ q^2_l = [\mu_l, W^2_x[x^1, x^2, \ldots, x^{t-1}], \mu_l] \]
\[ q^2_r = [\mu_r, W^2_x[x^n, x^{n-1}, \ldots, x^{t+1}], \mu_r] \]
Left and right fragments encoded

\[ u_l^2 = LSTM_{LR}^2 (q_{l(i)}^2), \quad (i = 1, \ldots, (t + 1)) \]
\[ u_r^2 = LSTM_{RL}^2 (q_{r(i)}^2), \quad (i = 1, \ldots, (n - t + 2)) \]

Bounded Values

\[ u_q = \tanh(\mathbf{W}_{uq}[u_l^1 | u_r^1 | u_l^2 | u_r^2]), \]

Trainable weights to combine four LSTMs outputs
Video Encoding

Video

Text Encoder

Video Encoder

Temporal Attention

Spatial Attention

Temporal Video Feature

Spatial Video Feature
Someone watches out of the corner of his eye as the kid finds a cheap _____ inside.

(Answer: Sweet )
Someone watches out of the corner of his eye as the kid finds a cheap _____ inside. (Answer: Sweet)
Spatial Attention Module

- Which **regions** of the frames to look?

- VGG-19 pre-trained network **last convolution** layer

  - $14 \times 14 \times 512$
  - **196** regions and **512** dimensional feature vectors

- Each region corresponds to a $32 \times 32$ **pixels** patch

Someone watches out of the corner of his eye as the kid finds a cheap _____ inside.  (Answer: Sweet)
Spatial Attention – Block Diagram

- **Source Sentence**
  - Textual Encoder
  - Augmentation

- **Video Frames**
  - VGG19
  - Max Pooling over last Pooling layers

- **Augmented Textual Features**
  - \([k \times m]\)

- **Visual Features**
  - \([k \times m]\)

- **Combined Representation**
  - \([k \times m]\)

- **Trainable Weights**
  - \([k \times 1]\)

- **Spatial Attention**
  - \([m]\)

m: Number of regions (196)
k: Textual and visual joint representation length
Spatial Attention Networks

\[ \Phi_F = \tanh(W_f \Theta(\Phi_f(f^t)|_{f^t \in F}) \]

\[ \Psi_F = \tanh((W_F \Phi_F) \oplus (W_u u_q + b_u)) \]
Spatial Attention Networks

\[ p_{sp} = \text{softmax}(\Psi_F^T w_{sp}), \]

\[ u_{sp} = \Phi_F p_{sp}, \]

Attention as probability

Weighted Average

Bounded Value
Temporal Attention Model

• C3D Features for each 16 frames
  • C3D is a temporal-convolutional network
    • Pre-trained on SPORT1M dataset
    • Encodes short shots (16 frames)

• Videos are longer than 16 frames
  • Sequence of 16 frames shots

• Which shot is more important?
Temporal Attention Network

Source Sentence
Textual Encoder

Video Shots
C3D

LSTM Encoder
LSTM

Attention Vectors
\( \Omega^1_G \) \( \Omega^2_G \) \( \ldots \) \( \Omega^L_G \)
Temporal Attention Network

\[ \Phi_G = \tanh(W_g[\Phi_g(g^1), \Phi_g(g^2), ..., \Phi_g(g^{|G|})]), \]

\[ \Psi_G = \tanh((W_G \Phi_G) \oplus (W_u u_q + b_u)), \]

\[ \Omega_G^i = \text{LSTM}(\Psi_G^i) \quad (i = 1...|G|), \]

\[ p_{tp} = \text{softmax}(\Omega_G^T w_{tp}), \]

\[ u_{tp} = \Phi_G p_{tp}, \]
Inference Module

\[ u = [u_q + u_{sp} + u_{tp}] , \]

\[ P_{blank} = softmax(W_{blank}u) , \]

\[ \hat{b} = \arg \max_{b \in \beta} P_{blank}(b) \]
Experiments

• Large Scale Movie Description Challenge 2016 (LSMDC16)
• About 360,000 samples
• Dataset is built upon movies audio descriptions
  • Complicated sentences
  • Large dictionary of words (21,000)
• Videos vary in length
  • 2 – 60 seconds
Training

- End to End training
  - All modules trained together
- Adagrad Optimizer
  - Any adaptive solver
- Cross-categorical Loss Function
- Batch-normalization before all non-linearity
- 8-10 hours training time on TitanX GPU
Quantitative Results

- Human Accuracy $\sim 0.68$
- Text Only Methods (Blind Test)
- Video Only (Lazy Student)
- Video + Text Methods
- Our Method

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Only</td>
<td>0.006</td>
</tr>
<tr>
<td>Random Guess</td>
<td>0.006</td>
</tr>
<tr>
<td>LSTM Left Sentence</td>
<td>0.155</td>
</tr>
<tr>
<td>LSTM Right Sentence</td>
<td>0.165</td>
</tr>
<tr>
<td>BiLSTM</td>
<td>0.320</td>
</tr>
<tr>
<td><strong>Our Sentence Encoding</strong></td>
<td><strong>0.367</strong></td>
</tr>
<tr>
<td><em>Human [27]</em></td>
<td>0.302</td>
</tr>
</tbody>
</table>
Someone grabs her arm, pulls her close and **gives** her a lingering kiss.

**Answer:** gives  
**GT:** gives
Someone watches out of the corner of his eye as the kid finds a cheap _____ inside.

Answer: Sweet

GT: Sweet

Qualitative Results
Qualitative Results

Someone defensively grabs a picture frame and presses ___ back to a wall by a white-trim doorway.

Answer: her

GT: her
Qualitative Results

Temporal Attention

Someone grabs her arm, pulls her close and ____ her a lingering kiss.

Spatial Attention

GT: gives  Ours: gives

Someone watches out of the corner of his eye as the kid finds a cheap ____ inside.

GT: sweet  Ours: sweet

Someone defensively grabs a picture frame and presses ___ back to a wall by a white-trim doorway

GT: her  Ours: her
Qualitative Results - Failures

Someone stops his ____ and kisses her on the head.

GT: daughter  
Ours: jacket
Fill in Multiple Blanks

• More than one blank in sentence
  • More applications
  • Higher level of alignment between text and video

• More Challenging problem!
  • Each left and right fragments can be fragmented by themselves
Someone ____ his serious ____ from the contract.

<table>
<thead>
<tr>
<th>Left Fragment</th>
<th>Right Fragment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blank 1</td>
<td>Someone</td>
</tr>
<tr>
<td>Blank 2</td>
<td>His serious</td>
</tr>
<tr>
<td></td>
<td>From the contract</td>
</tr>
</tbody>
</table>

Subdivision(cutting) method

<table>
<thead>
<tr>
<th>Left Fragment</th>
<th>Right Fragment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blank 1</td>
<td>Someone</td>
</tr>
<tr>
<td>Blank 2</td>
<td>Someone his serious</td>
</tr>
<tr>
<td></td>
<td>From the contract</td>
</tr>
</tbody>
</table>

Masking method

LR/RL LSTMs with Spatial and Temporal Attention Models

Blank 1 = **Lifts**  
Blank 2 = **Gaze**
## Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baselines</strong></td>
<td></td>
</tr>
<tr>
<td>Random Guess</td>
<td>0.006</td>
</tr>
<tr>
<td>Left LSTM (Masking)</td>
<td>0.104</td>
</tr>
<tr>
<td>Bi-LSTM (Masking)</td>
<td>0.156</td>
</tr>
<tr>
<td>2Videos + Textual (Subdivision) [14]</td>
<td>0.136</td>
</tr>
<tr>
<td>2Videos + Textual (Masking) [14]</td>
<td>0.177</td>
</tr>
</tbody>
</table>
Someone _____ and _____ his ______.

Cutting:
2Videos – Textual Encoding (turns, faces, gaze)
Our Model (smiles, raises, wife)

Masking:
2Videos – Textual Encoding (smiles, shakes, gaze)
Our Model (smiles, shakes, head)
Ground Truth: (smiles, shakes, head)
She ______ off _______ shoes and _______ them aside.

**Cutting:**
- 2Videos – Textual Encoding:  
  - (walks, her, sets)
- Our Model:  
  - (tosses, her, sets)

**Masking:**
- 2Videos – Textual Encoding:  
  - (takes, her, sets)
- Our Model:  
  - (takes, her, tosses)
- Ground Truth:  
  - (takes, her, flings)
Conclusion

• Discussed the VFIB problem
• Novel sequence encoder for fragmented inputs
• Detailed formulation for spatial and temporal attention
• Quantitative and Qualitative Results
• Multiple Blanks cases
Video Fill In the Blank using LR/RL LSTMs with Spatial-Temporal Attentions

Amir Mazaheri, Dong Zhang, and Mubarak Shah
ICCV 2017
http://crcv.ucf.edu/papers/iccv17/PID4929115.pdf
Contents

- Sematic Segmentation
- Facial Attributes Detection
- Human Re-Identification
- Target Detection in WAMI
- Anomaly Detection
- Human Action Localization
- Video Fill In The Blank
- Reading The Mind
Deep Learning Human Mind for Automated Visual Classification

Concetto Spampinato, Simone Palazzo, Isaak Kavasidis, Daniela Giordano
PeRCeiVe Lab, University of Catania, Italy
Nasim Souly, Mubarak Shah
CRCV, University of Central Florida, USA
CVPR 2017

Motivation

• CNNs emulate human visual cortex

• What do CNNs miss?

http://wiki.bethancrane.com/introducingtheeye
• Visual classification in humans stands at the interface between **perception** (visual cortex) and **conception** (cognitive processes)
Harnessing visual-cognitive factors for visual categorization

Brain activity in EEG recordings contain information about visual object categories [1, 2].

Dataset

• 6 subjects, asked to look at pictures while recording EEG
• 40 ImageNet classes
• Image duration 500 ms
• 128 channel EEG, 1000 Hz sampling rate, 16 bit resolution
• Dataset available:
  • http://perceive.dieei.unict.it/files/eeg_data_cvpr_2017.zip
Dataset: 40 ImageNet Categories
• Can class-discriminative EEG features be extracted from raw EEG signals?
Approach
Human-Machine Computer Vision Systems

Bottom-up approach: brain-driven visual classifier

EEG features can be used for automated image classification
EEG Feature Encoding

Stacked LSTMs

Common

Channel + Common

Common + Output
# EEG Classification Accuracy

<table>
<thead>
<tr>
<th>Model</th>
<th>Details</th>
<th>Max VA</th>
<th>TA at max VA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common</td>
<td>64 common</td>
<td>74.4%</td>
<td>73.9%</td>
</tr>
<tr>
<td></td>
<td>128 common</td>
<td>77.3%</td>
<td>74.1%</td>
</tr>
<tr>
<td></td>
<td>64,64 common</td>
<td>75.9%</td>
<td>72.5%</td>
</tr>
<tr>
<td></td>
<td>128,64 common</td>
<td>79.1%</td>
<td>76.8%</td>
</tr>
<tr>
<td></td>
<td>128,128 common</td>
<td>79.7%</td>
<td>78.0%</td>
</tr>
</tbody>
</table>
Deep Learning Human Mind for Automated Visual Classification

Results: **Investigation of Cognitive Factors**

- 50 – 120 ms: Feature extraction for object recognition [1]
- >120 ms: ???

<table>
<thead>
<tr>
<th>Visualization time</th>
<th>Max VA</th>
<th>TA at max VA</th>
</tr>
</thead>
<tbody>
<tr>
<td>40-480 ms</td>
<td>85.4%</td>
<td>82.9%</td>
</tr>
<tr>
<td>40-160 ms</td>
<td>81.4%</td>
<td>77.5%</td>
</tr>
<tr>
<td>40-320 ms</td>
<td>82.6%</td>
<td>79.7%</td>
</tr>
<tr>
<td>320-480 ms</td>
<td>86.9%</td>
<td>84.0%</td>
</tr>
</tbody>
</table>

## Previous Methods

<table>
<thead>
<tr>
<th>Work</th>
<th>Method</th>
<th>Object</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bashivan et al. [1]</td>
<td>Combination of CNN and RNN</td>
<td>Cognitive load classification</td>
<td>90% accuracy over 4 classes</td>
</tr>
<tr>
<td>Stober et al. [2]</td>
<td>CNNs</td>
<td>Classification of EEG signals evoked by songs</td>
<td>28% accuracy over 12 classes</td>
</tr>
<tr>
<td>Kaneshiro et al. [3]</td>
<td>SVM</td>
<td>Classification of brain signals evoked by visual stimuli</td>
<td>29% accuracy over 12 classes</td>
</tr>
</tbody>
</table>

---


Can EEG features be used for automated image classification?
Network
Human-Machine Computer Vision Systems

EEG feature regression architecture

Deep network used for visual feature extraction

Regression network trained over EEG features

EEG feature representation
- Regression accuracy:
  - Mean square error of regressed EEG features
  - Training: each image’s target was the average EEG vector over the 6 subject

<table>
<thead>
<tr>
<th>Feature set</th>
<th>AlexNet</th>
<th>GoogleNet</th>
<th>VGG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>k-NN</td>
<td>Ridge</td>
<td>RF</td>
</tr>
<tr>
<td>Average</td>
<td>1.64</td>
<td>1.53</td>
<td>1.52</td>
</tr>
<tr>
<td></td>
<td>k-NN</td>
<td>Ridge</td>
<td>RF</td>
</tr>
<tr>
<td></td>
<td>0.62</td>
<td>1.88</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>k-NN</td>
<td>Ridge</td>
<td>RF</td>
</tr>
<tr>
<td></td>
<td>0.73</td>
<td>1.53</td>
<td>0.94</td>
</tr>
</tbody>
</table>
CALTECH 101 (30 classes)
RESULTS

<table>
<thead>
<tr>
<th></th>
<th>GoogleNet</th>
<th>VGG</th>
<th>Our method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>92.6%</td>
<td>80.0%</td>
<td>89.7%</td>
</tr>
</tbody>
</table>
Video Fill In the Blank using LR/RL LSTMs with Spatial-Temporal Attentions

Amir Mazaheri, Dong Zhang, and Mubarak Shah
ICCV 2017
http://crcv.ucf.edu/papers/iccv17/PID4929115.pdf
Generative Adversarial Networks Conditioned by Brain Signals

S. Palazzo, C. Spampinato, I. Kavasidis, D. Giordano
PeRCeiVe Lab, University of Catania, Italy
www.perceivelab.com

M. Shah
CRCV, University of Central Florida, USA
http://crcv.ucf.edu

ICCV-2017

http://crcv.ucf.edu/papers/iccv17/egpaper_for_review.pdf
Generative models to generate images from brain signals
GAN (Generative Adversarial Network)
GAN (Generative Adversarial Network)

\[ \mathcal{L}_D = -\log D(x_t|y_t) - \log (1 - D(x_c|y_w)) - \log (1 - D(x_w|y_w)) \]
Human-Machine Computer Vision Systems

Bottom-up approach: brain-driven visual classifier

\[ \mathcal{L}_G = - \log D(x_w | y_w) \]

\[ \mathcal{L}_D = - \log D(x_c | y_c) - \log (1 - D(x_c | y_w)) - \log (1 - D(x_w | y_w)) \]

S. Palazzo, C. Spampinato, I. Kavasidis, M. Shah, “Generative Adversarial Networks Conditioned by Brain Signals” ICCV 2017
Generated Images

(a) Airliner

(b) Jack-o’-Lantern
Figure 3. Good results

(c) Panda

(a) Banana

(b) Capuchin
Figure 4. Bad results

(c) Bolete
Generative Adversarial Networks Conditioned by Brain Signals

S. Palazzo, C. Spampinato, I. Kavasidis, D. Giordano
PeRCeVe Lab, University of Catania, Italy
www.perceivelab.com

M. Shah
CRCV, University of Central Florida, USA
http://crcv.ucf.edu

ICCV-2017

http://crcv.ucf.edu/papers/iccv17/egpaper_for_review.pdf
Contents

- Target Detection in WAMI
- Facial Attributes Detection
- Human Re-Identification
- Semantic Segmentation
- Human Action Localization
- Video Fill In The Blank
- Reading The Mind
Contents

• PART-I: Deep Learning: A Short Overview

• PART II: Computer Vision Employing Deep Learning
Conclusions

• Deep Learning has been disruptive to Computer Vision
• Deep Computer Vision is being used
  • Self Driving Cars
  • Robotics
  • Health Care
  • Language and Vision
  • Sound and Vision
• Artificial General Intelligence
  • Alpha Go (zero)