

# In-vehicle baby alert system

## Advanced Digital Image Processing project

F. Casciola, E. G. Ceroni, N. Landolfi

Università degli Studi di Siena

date TBD

# Introduction

Vehicular heatstroke is largely underestimated by the general public. The majority of parents are misinformed and likely to believe that they could **never forget** their child in a vehicle.

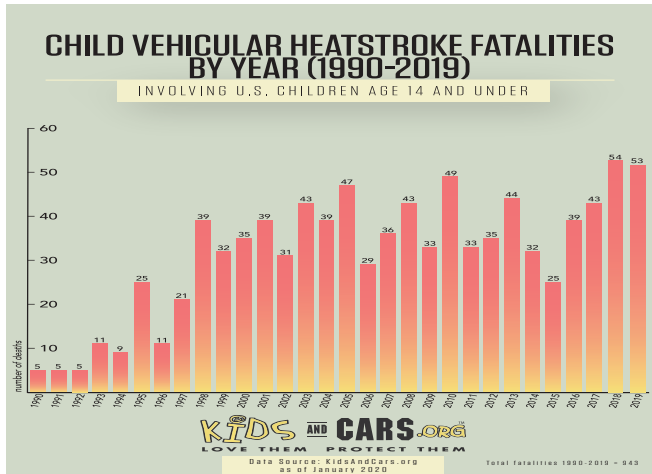
In over 55% of these cases, the person responsible for the child's death unknowingly left them in the vehicle. The most dangerous mistake one can make is to think leaving a child alone in a vehicle could never happen to them.

# Introduction

The inside of a vehicle heats up very quickly! Even with the windows cracked, the temperature inside a car can reach **51 degrees Celsius** in minutes.

A child's body overheats three to five times faster compared to an adult, and heatstroke occurs when the body's temperature exceeds 40 degrees Celsius and the body organs begin to shut down.

# Introduction: Some Data



# Introduction: Project Proposal

Based on what we have learned about Computer Vision and Image Processing, a possible solution would be to design a new system which enables **adult/child's face detection**.

Even better, a hybrid solution which combines several way of measuring/sensing the child's presence would be more robust.

Our proposal is composed of three main steps:

- Collecting the data and building a dataset;
- Model selection and synthetic testing;
- Field testing of the best model.

# The Dataset: Collecting The Data

This is the most challenging part of the project. Getting pictures of children under the age of 3 years old is not that easy.

In the beginning, we scraped images from Google Images, but we opted for a pre-existing licensed dataset<sup>1</sup>.

---

<sup>1</sup>Eran Eiding, Roeen Enbar, and Tal Hassner. “Age and gender estimation of unfiltered faces”. In: *IEEE Transactions on Information Forensics and Security* 9.12 (2014), pp. 2170–2179.

# The Dataset: Sub-sampling And Dataset Adjustments

Since the pre-existing dataset is designed for a multi-class age classification task, we applied sub-sampling.

This yields an equal number of samples for adults and children, thus focusing the problem on a **binary classification task**.

Moreover, we decided that the images should mostly contain faces with as little background as possible. To this end, we fed our images into a face extractor<sup>2</sup>.

---

<sup>2</sup>We settled for MTCNN over HAAR cascade.

# Dataset - Final

Our final dataset has been split in:

- Training set: 3520 child faces and 3624 adult faces
- Validation set: 379 child faces and 401 adult faces
- Test set: 387 child faces and 238 adult faces



# Face extractor

As mentioned above, we used a face extractor for two reasons:

- Training set creation: labeling faces by hand was too slow and tedious
- Extraction of faces from the acquired image (main use case)

We began with HAAR cascade, both frontal and lateral, then switched to MTCNN, which proved far superior.

# MTCNN

## Framework:

- Image resizing for the creation of a pyramid of images
- The image pyramid is fed to three different CNNs:
  - 1 First, the **P-Net** produces a large number of candidate BBs<sup>3</sup> and performs BB regression, followed by NMS<sup>4</sup> for merging the overlapping ones.
  - 2 Surviving candidates are fed to the **R-net** that performs BB regression and again NMS.
  - 3 At last, the survived boxes are fed to the **Q-net** that performs similarly to the R-net but it is more complicated and outputs the positions of **five facial landmarks**.

---

<sup>3</sup>BB = Bounding box

<sup>4</sup>Non-maximum suppression

# MTCNN

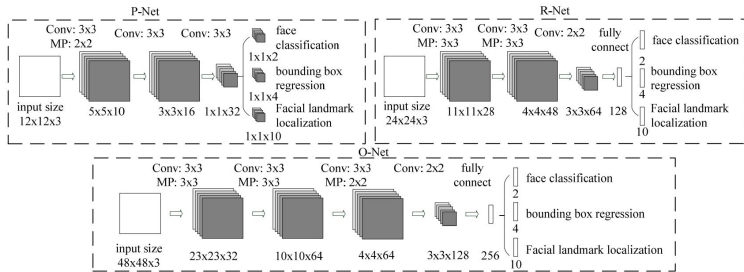


Fig. 2. The architectures of P-Net, R-Net, and O-Net, where “MP” means max pooling and “Conv” means convolution. The step size in convolution and pooling is 1 and 2, respectively.

Figure: MTCNN schematics<sup>5</sup>.

<sup>5</sup>Kaipeng Zhang et al. “Joint face detection and alignment using multitask cascaded convolutional networks”. In: *IEEE Signal Processing Letters* 23.10 (2016), pp. 1499–1503.

# Fischerface - Generalities

TODO

# Siamese Neural Network

As previously mentioned, age classification is a challenging problem due to the complexity of the features that make up a face.

So we chose a **discriminative** approach, since we want to be able to separate **children** from **non-children**.

This was achieved by taking advantage of a **Siamese neural network**<sup>6</sup> that takes two inputs: a **template** image and the input image from the face extractor and checks if they belong to the same class or not.

---

<sup>6</sup>Actually there is only one network that is used to process the two inputs.

# Siamese Neural Network - Template/Example pairs

This kind of neural networks require in input a couple of images:

- Template image: the class example
- Input image: the image that has to be classified

We used  $(1, 0)$  as the label if the template image and the input image belonged to the same class,  $(0, 1)$  otherwise, given that this is a **binary classification problem**.

## Siamese Neural Network - Template/Example pairs

We selected 26 child images as templates, and paired them with all the other images in the original dataset <sup>7</sup>, obtaining a new larger set of examples.

### Creating Datasets

#### Training set:

Number of same class image pairs = 95391, Number of different class image pairs = 97848, total sample pairs: 193239

#### Validation set:

Number of same class image pairs = 10584, Number of different class image pairs = 10827, total sample pairs: 21411

#### Test set:

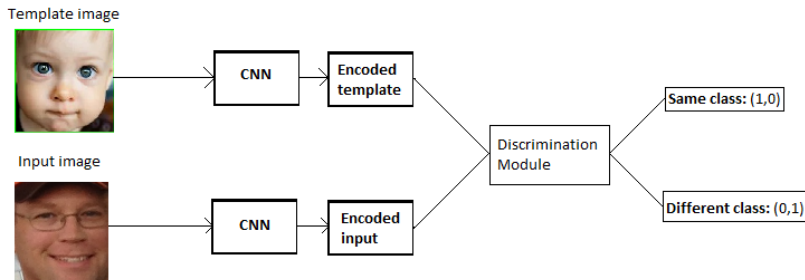
Number of same class image pairs = 10800, Number of different class image pairs = 6426, total sample pairs: 17226

Figure: Siamese training - validation - test set (children network)

---

<sup>7</sup>We excluded the pairs which contained the same image

# Siamese Neural Network - General Architecture



**Figure:** General outline of the network, the CNN is the same for both images



## Siamese Neural Network - Discrimination module

The CNN part of the system actually works as an image encoder, extracting features from both the template and the input image <sup>8</sup>, which are then fed to the **discrimination module**.

The paper<sup>9</sup> that inspired this approach used a joining neuron that calculated the cosine distance between the encoded vectors.

We decided to implement two different discriminator modules, one based on the **euclidean distance** between the CNN-encoded vectors and for the other one a **multi-layered perceptron** which was fed the concatenation of the two encoded vectors.

---

<sup>8</sup>Could be optimized at runtime by preprocessing the templates

<sup>9</sup>Jane Bromley et al. "Signature verification using a" siamese" time delay neural network". In: *Advances in neural information processing systems*. 1994, pp. 737–744.

# Siamese Neural Network - Model selection strategies

We began our work by implementing a modified, slimmed-down version of the VGG16 architecture, based on the remarkable results that this model obtained in **ILSVRC**<sup>10</sup> 2014.

However, we were not satisfied with the results, so we decided to build a custom network and started the cross-validation process. This however was taking too long even for a small subset of hyperparameters (although it was giving very decent results when we stopped it, see table below).

So we devised a very simple evolutionary algorithm, with accuracy on validation set as fitness function.

---

<sup>10</sup>ImageNet Large Scale Visual Recognition Competition

# Use-case process pipeline

The overall child recognition pipeline consists in **3 steps**:

- 1 Acquisition of image from image sensor
- 2 Extraction of the faces in the image with MTCNN
- 3 Face classification using the ensemble

# Fisherface - Training results

TODO

## Siamese Neural Network - Training results

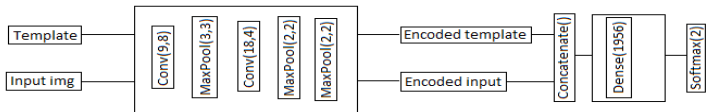
We tested the resulting networks on the appropriately<sup>11</sup> held-out test set and obtained<sup>12</sup>

Model	Accuracy	Loss
VGG16 + MLP (Adults)	90.43%	0.5991 (H)
VGG16 + MLP (Child)	90.24%	0.5986 (H)
CV (Child)	93.06%	0.5751 (H)
<b>Evo (Child)</b>	<b>94.07%</b>	<b>0.5628 (H)</b>
<b>Evo (Adults)</b>	<b>93.46%</b>	<b>0.5658 (H)</b>

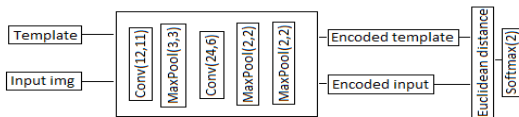
<sup>11</sup>Two different sets, one for adults and one for children

<sup>12</sup>H = Hinge loss

# Siamese Neural Network - Selected architectures



Network architecture (adults): lr = 0.0001, Activation (CNN): relu, Activation (MLP): tanh, Loss: Hinge



Network architecture (children): lr = 0.00026092515297289765, Activation(CNN) = Elu, Loss = Hinge

Figure: Selected architectures, Optimizer (both): Nadam

# Autoencoder

TO BE DETERMINED

# SIFT-SURF trade-off

TODO trade-off tra dimensione del set di indicatori e velocità di esecuzione



# RGB vs BGR in OpenCV

TODO

# Min face dimension for MTCNN

TODO

*Thank You.*