Rock, Paper & Scissors!

Silvia-Laura Pintea [6109969]

Nimrod Raiman [0336696]

Task Description

Teach *Nao* how to play **"Rock, Paper & Scissors"**

- Extract hands from webcam stream
- Learn a model for hand-gestures of: rock, paper and scissors
- Classify extracted hands
- Make *Nao* generate the moves for *"rock"*, *"paper"* and *"scissors"*
- Make *Nao* keep the score of the game by recognizing the gestures of the other player

Our approach

1. Extract hands from webcam stream
   - Detect the face of the player
   - Get the probability histogram
   - Backproject probability histogram on image frame
   - Extract the part corresponding to a hand
2. Recognize the gestures
   - Build reliable training set
   - Find useful features to train on: *PCA*, *Gabor wavelets*, *grayscale* images
   - Train a classifier: *Knn* / *SVM*
   - Create models & test in order to find the best one
3. Implement the motion and communication on *Nao*

Extracting hand from the frames

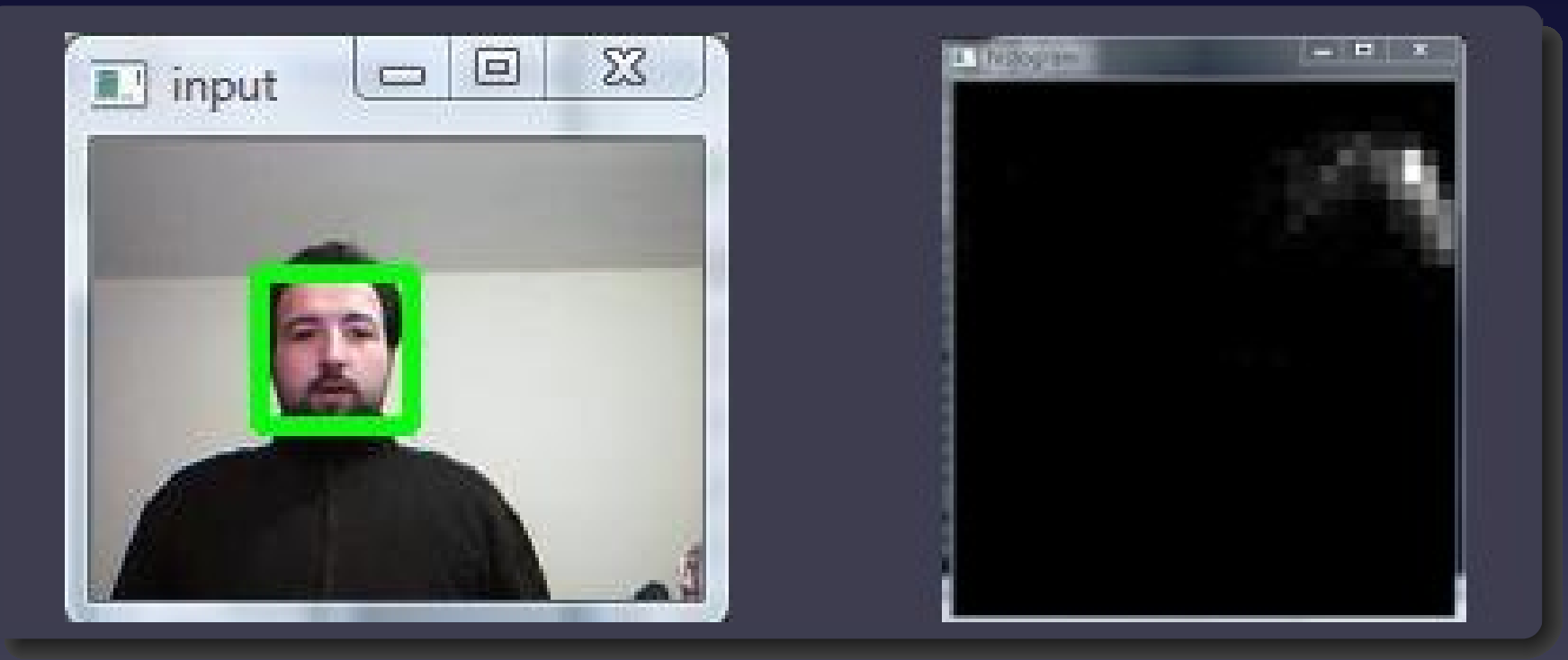
Naive approach (as shown in the majority of papers):

- Determine hue and saturation values for skin color
- Threshold the image with hue and saturation values
- Use erosion & dilation
- Find ares corresponding to hands

This approach is very sensitive for background noise.
Advice: Do not try this!

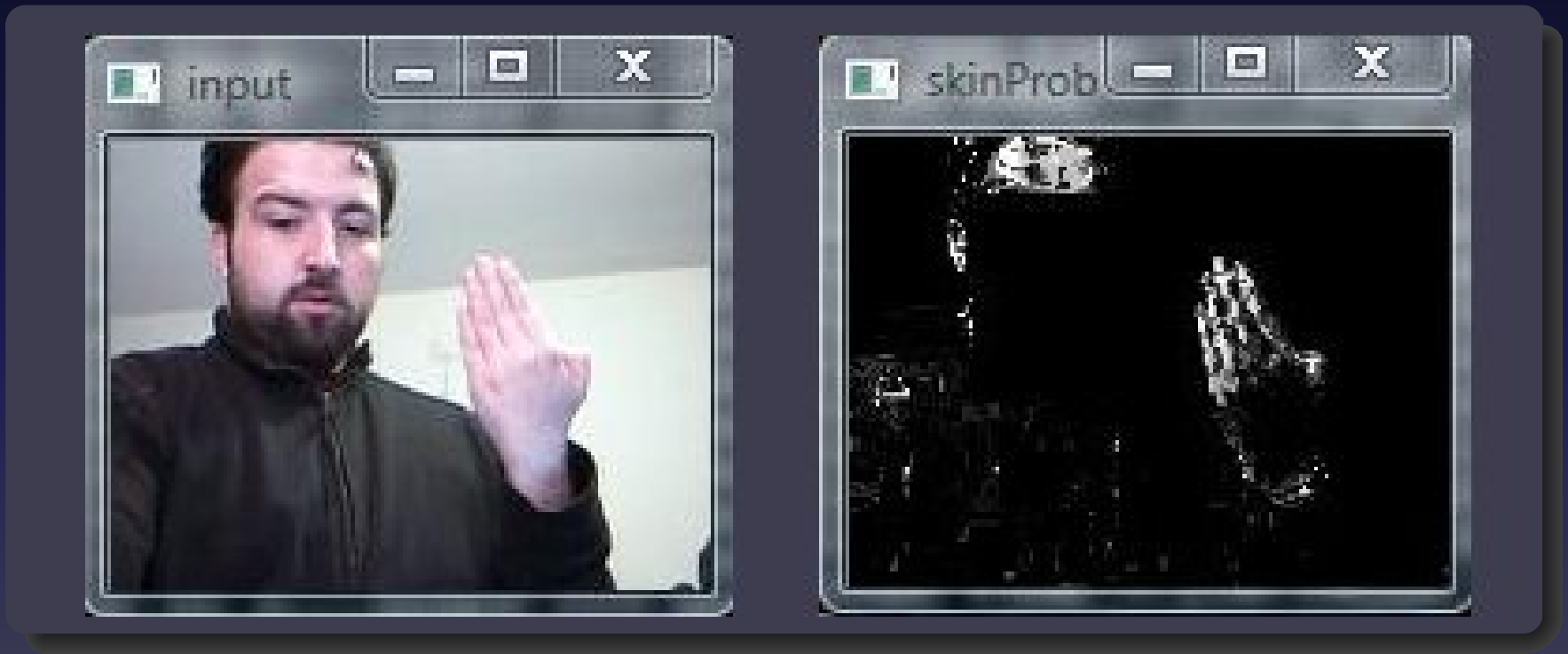
Extracting hand from the frames: An example

- Determine skin color histogram
  - Detect face
  - Build histogram of pixels corresponding to the face



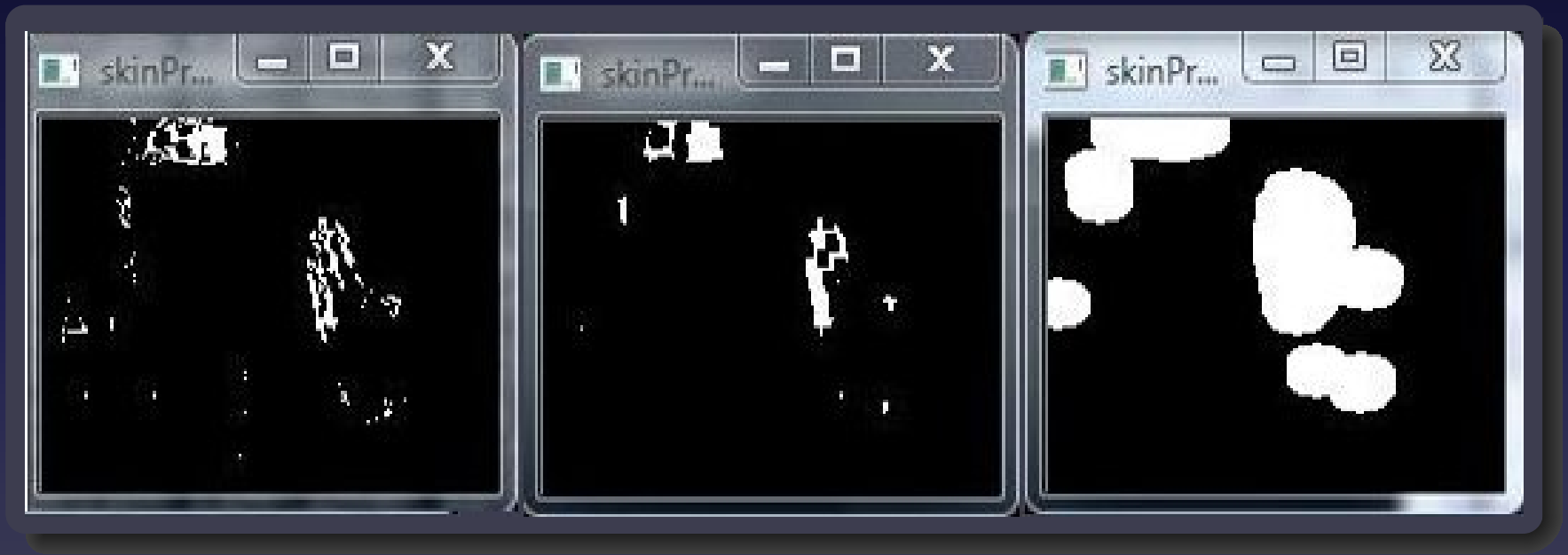
Extracting hand from the frames: An example

- Backproject skin color histogram on whole frame



Extracting hand from the frames: An example

- Use erosion & dilation to reduce the noise and fill up gaps
- Extract area of corresponding to the hand



Extracting hand from the frames: An example

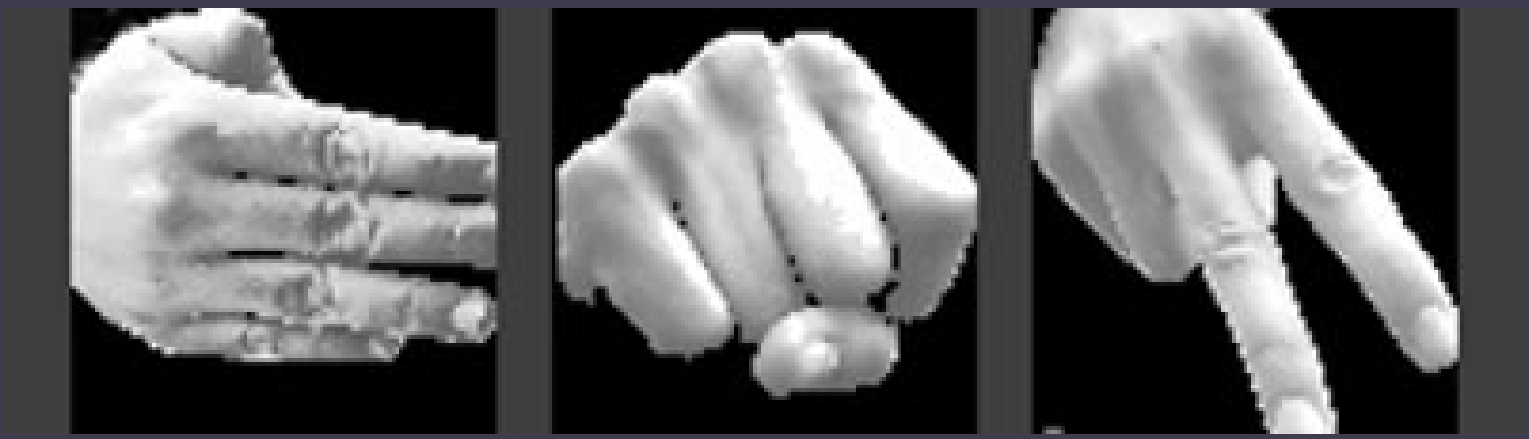
- Use more sophisticated erosion & dilation on hand area
  - Retain the hand and remove the background
  - Resize the area of interest to 70x70

Training Data

We have used:

Grayscale images of *70*×*70px* with hands at different angles and in different positions and with a black background.

Data – Example



Approximatively 1400 train images per sign.

Training Features – PCA

The steps for creating the *eigen-hands* are:

1. Extract the mean of the data
2. Compute the covariance of the data and the eigenvectors of the covariance matrix. For high-dimensional data compute:

$\mathbf{V} \rightarrow$ the eigenvectors of: $eigh(Data \times Data^T)$

$\mathbf{U} \rightarrow$ the final eigenvectors: $\frac{Data^T \times V}{norm(Data^T \times V)}$

3. Project each training set separately on the eigen-space

Training Features – Gabor wavelets

Gabor wavelets – Example

Train Features – Gabor wavelets

The steps for extracting the features:

1. Create the *Gabor wavelet* using the formula:

$$\mathbf{g(x,y,}\lambda\mathbf{,}\theta\mathbf{,}\psi\mathbf{,}\sigma\mathbf{,}\gamma\mathbf{)} = \exp\left(-\frac{x\prime^2+\gamma^2 y\prime^2}{2\sigma^2}\right)\cos\left(2\Pi\frac{x\prime}{\lambda}+\psi\right)$$

$$x\prime = \text{x } \cos\theta + \text{y } \sin\theta$$

$$y\prime = \text{-x } \sin\theta + \text{y } \cos\theta$$

2. Convolve the each image with the resulted wavelet
3. Reshape the convolved images on a single row (for multiple *Gabor wavelets* concatenate them)

Experimental Set-up

For gesture recognition we have tried using both the **SVM** and **Knn** classifiers.

Given the fact that the data is not aligned (hands have slightly different positions in the image and different angles) the problem seems to be too hard for the **SVM**.

In conclusion we have decided to use **Knn** for the classification.

We have used:

- *1641* training examples for the *"rock"* sign
- *1522* training examples for the *"paper"* sign
- *1377* training examples for the *"scissors"* sign

Evaluation Results

Average errors for each method

Size	Method	Rock	Paper	Scissors	Total
70×70	*PCA*	0.375	0.464	0.604	0.475
20×20	*PCA*	0.381	0.477	0.570	0.470
20×20	*Gabor*	0.017	0.010	0.039	0.021
20×20	*Gabor + PCA*	0.446	0.516	0.577	0.510
20×20	**Gabor & Image**	**0.008**	**0.005**	**0.022**	**0.012**
20×20	*(Gabor & Image) + PCA*	0.331	0.483	0.549	0.447
70×70	*Grayscale*	0.008	0.012	0.029	0.016
20×20	*Grayscale*	0.008	0.007	0.026	0.014

Conclusions

- SVM can handle only very simple cases where only one orientation per gesture is allowed
- PCA was not strong enough to extract features when all orientations for gestures are allowed
- KNN works best. Even on raw images
- When not perfect images are used (like the ones extracted by our skin-detector) results tend to drop
- More reliable results were obtained by enriching our training sets with examples of hands extracted by skin-detection with the method described above

Demo!

The third part of the project was to implement movement and communication on *Nao*.

- *Nao* is sweet (naknak)
- It has 500MhZ processor
- It has 3 fingers that move simultaneously
- Broke his hip 2 times in 6 month
- It costs 17.500 dollar (16.800k if you buy 5)
- His/her favorite word is *"heat!"*