# Machine Learning – How it works

Cat

Burglar breaking in a house

How does my computer do it ???

# Machine Learning – How it works

- Traditional Software Development:
  Software behavior determined by

  Logic written by developpers

- Machine Learning:
  Software behavior determined by

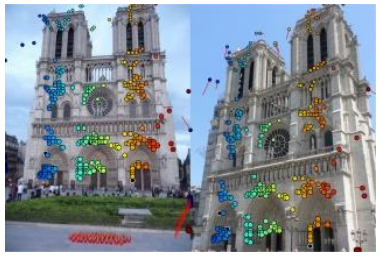
  **Data/examples**

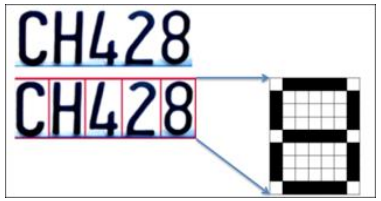
# Machine Learning – How it works

Convert RGB pixels to meaningful numbers that can be understood by the machine (hand-crafted features)


=> Feature Extraction

- Colors
- Shapes
- Points of interest
- Motion (video)
- Text/Numbers

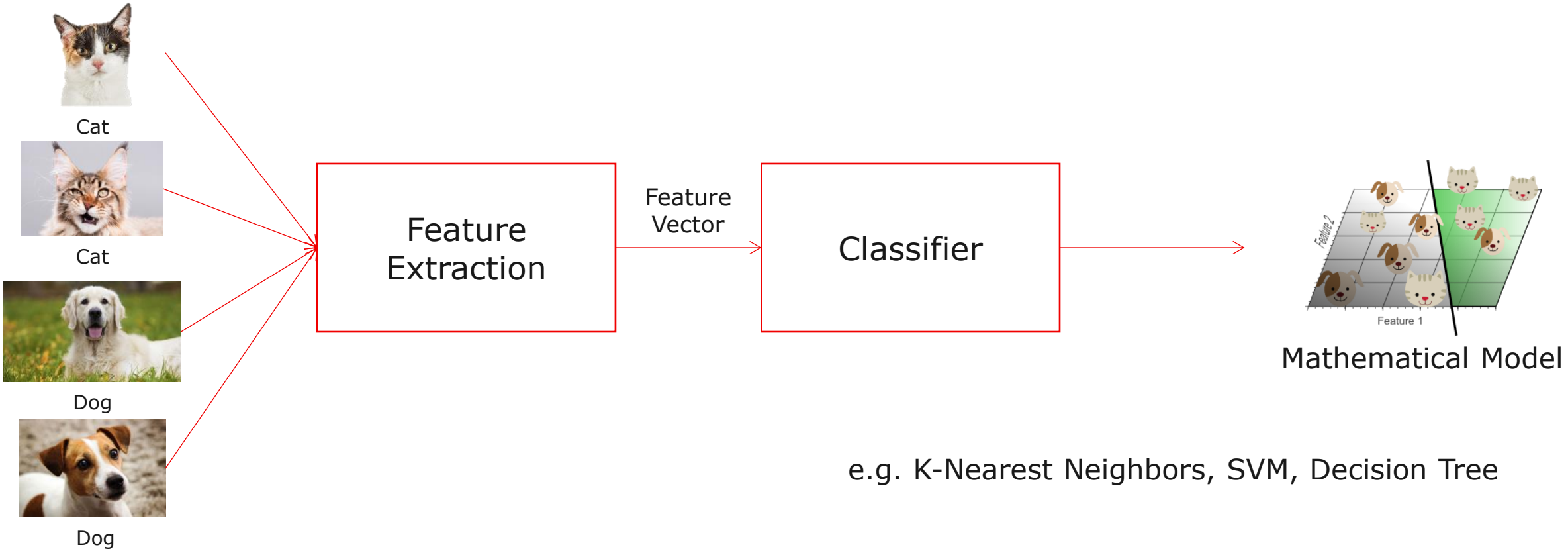





**Note:** We can also build hand-crafted features for audio but that is not in the scope of this track

# Machine Learning – How it works

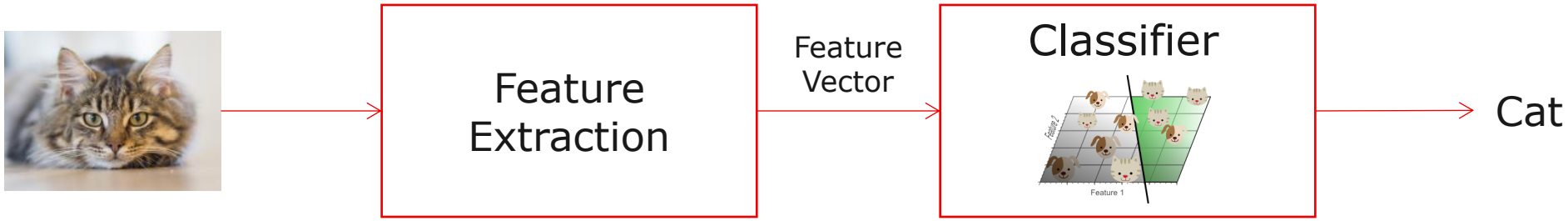
From meaningful features, we can learn patterns !
We feed what we call a "Classifier" with labelled data and it will build a mathematical model (Optimization). This is the training step.




e.g. K-Nearest Neighbors, SVM, Decision Tree

# Machine Learning – How it works

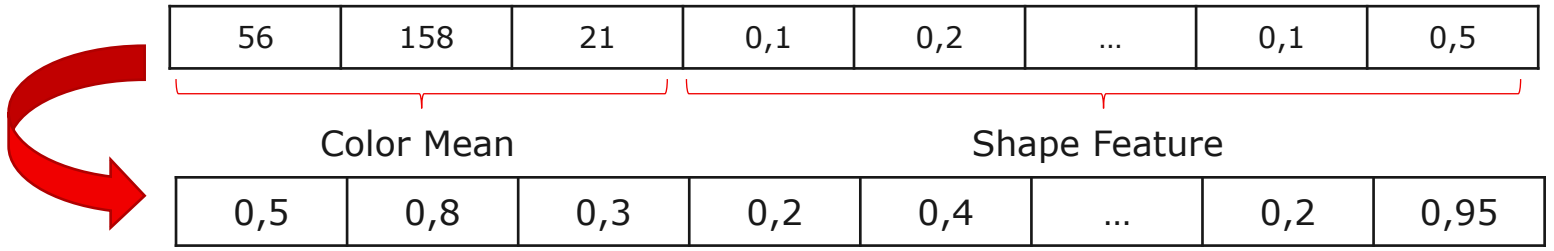
Once the classifier is trained (optimized mathematical model), we can
Feed it with new samples and predict its label.



# Machine Learning – How it works

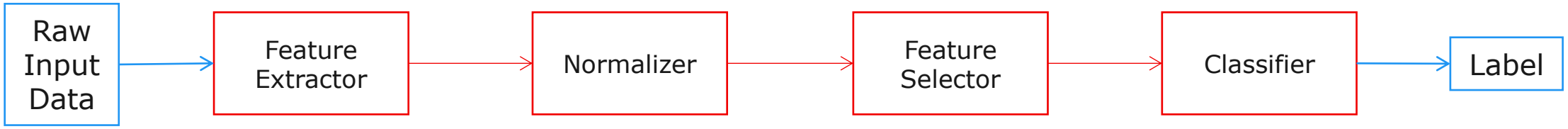
This pipeline is usually too simplistic to perform well

- Reducing the feature values is good for speed
- When concatenating feature vectors, some features can have larger values than others

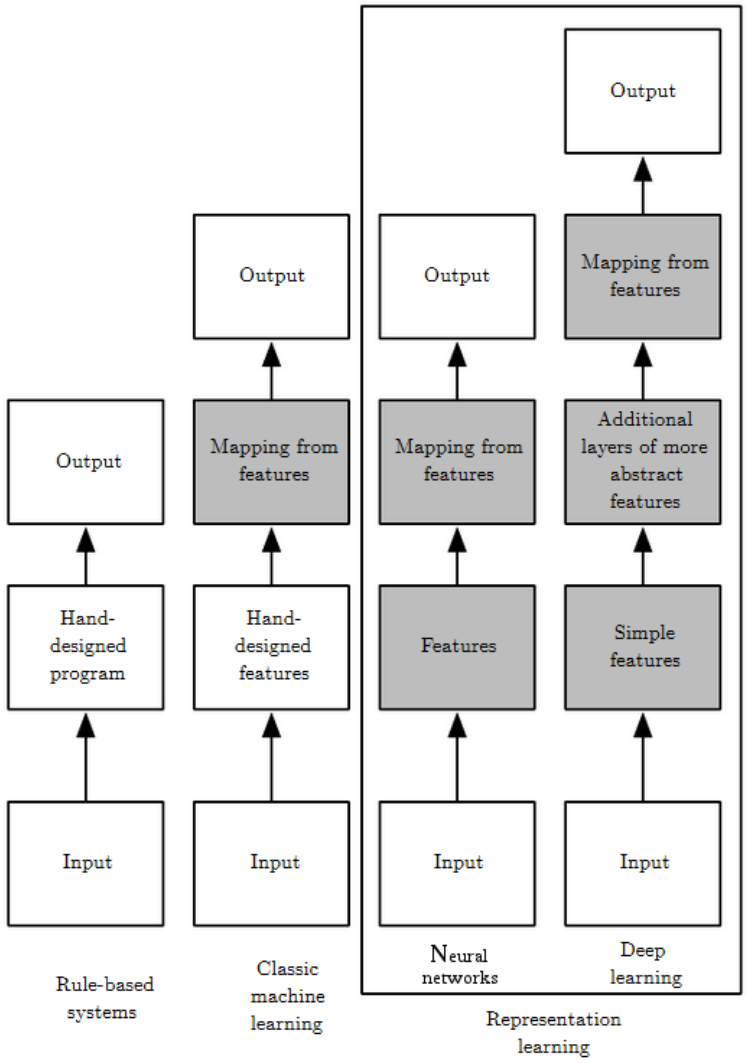


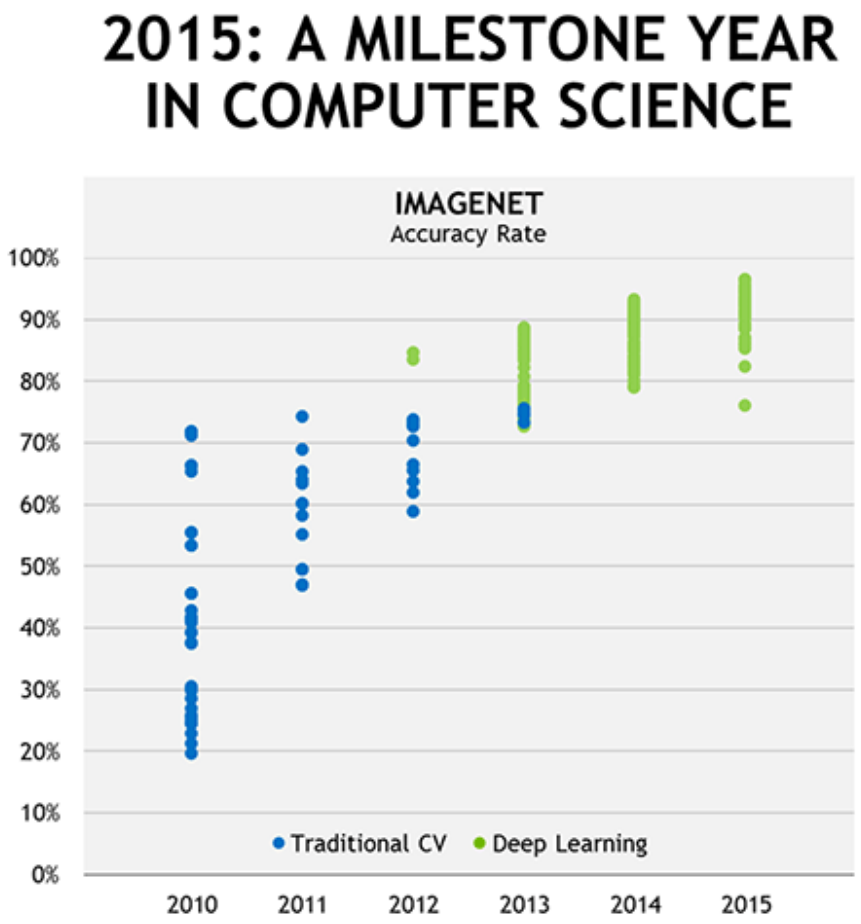

=> Normalize the feature vectors

- Select the relevant features only! E.g. Principal Component Analysis

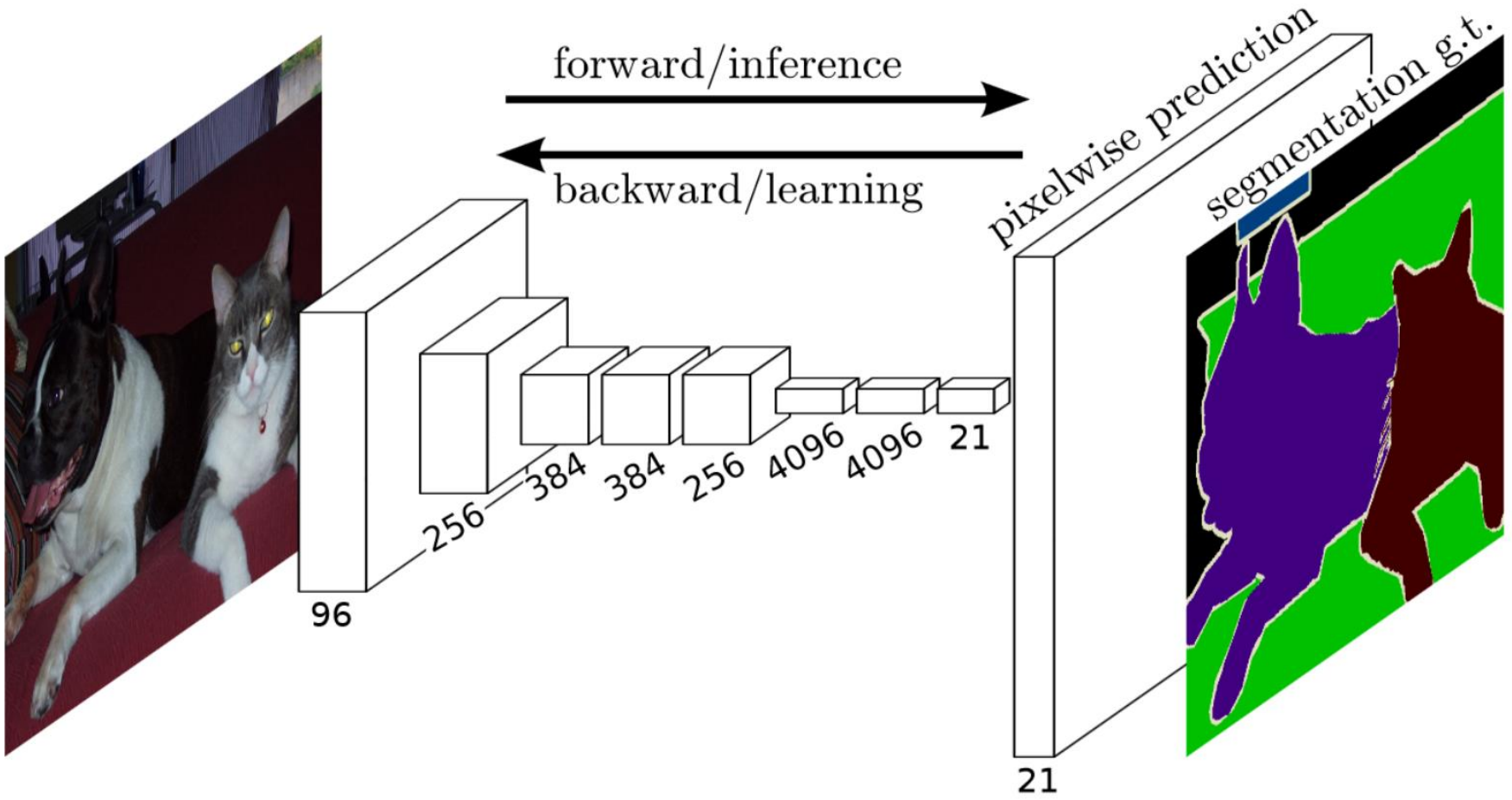


# Machine Learning – How it works

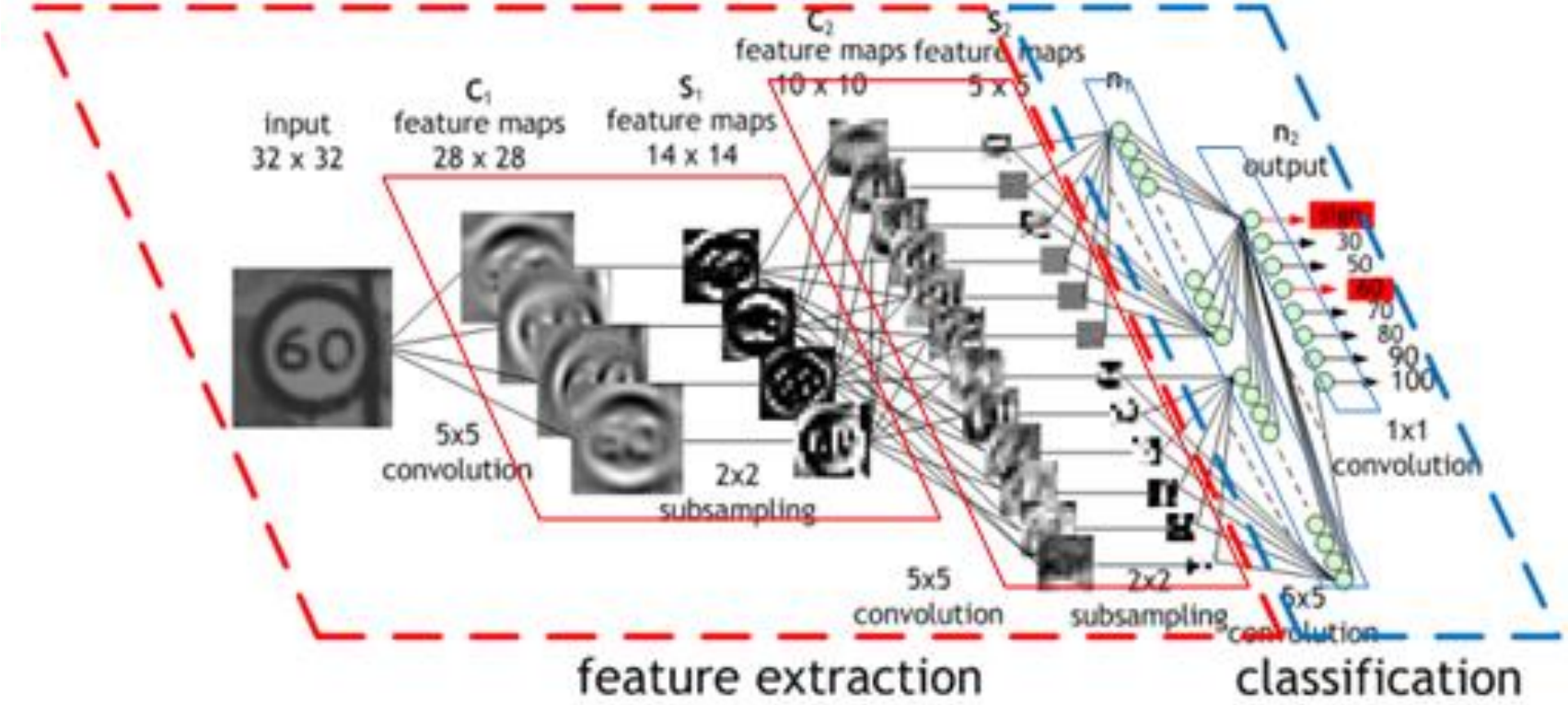


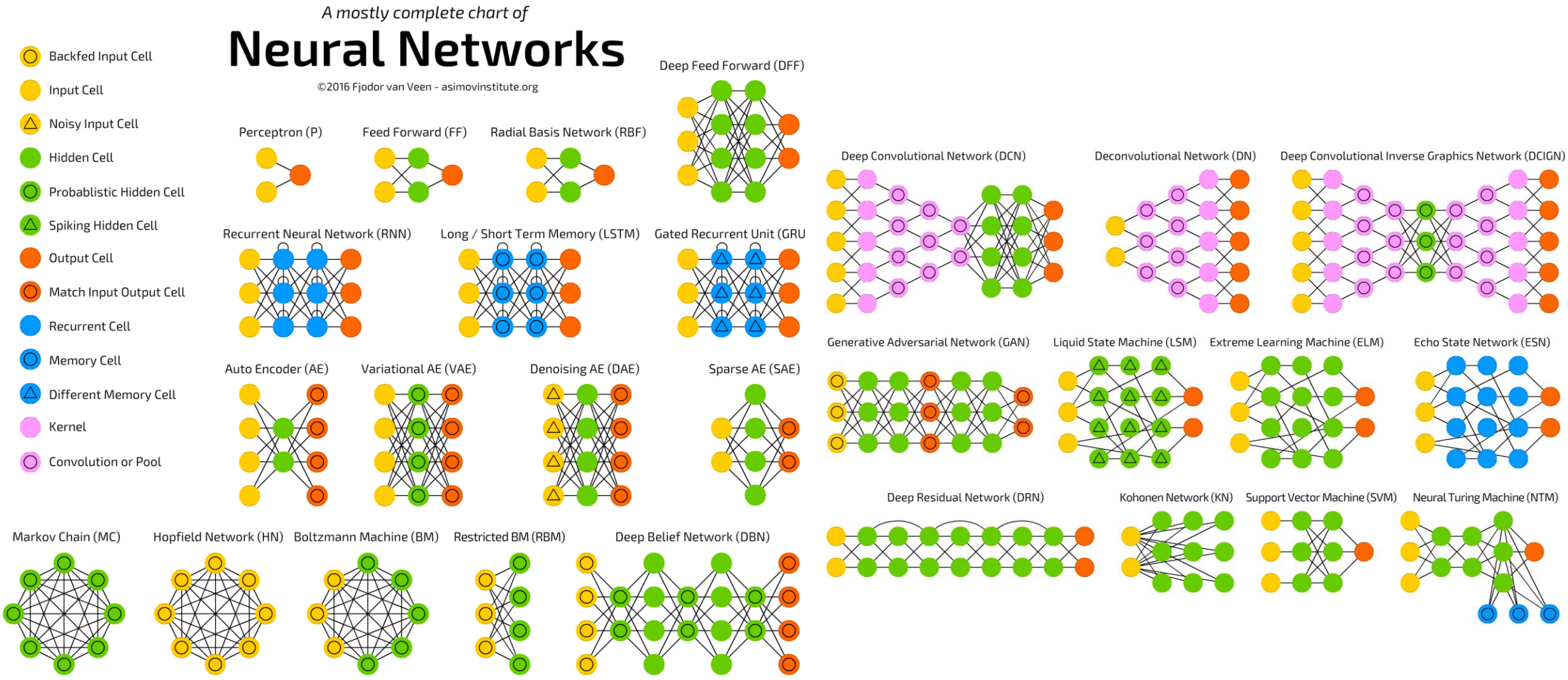
# Machine Learning – How it works



# Machine Learning – How it works

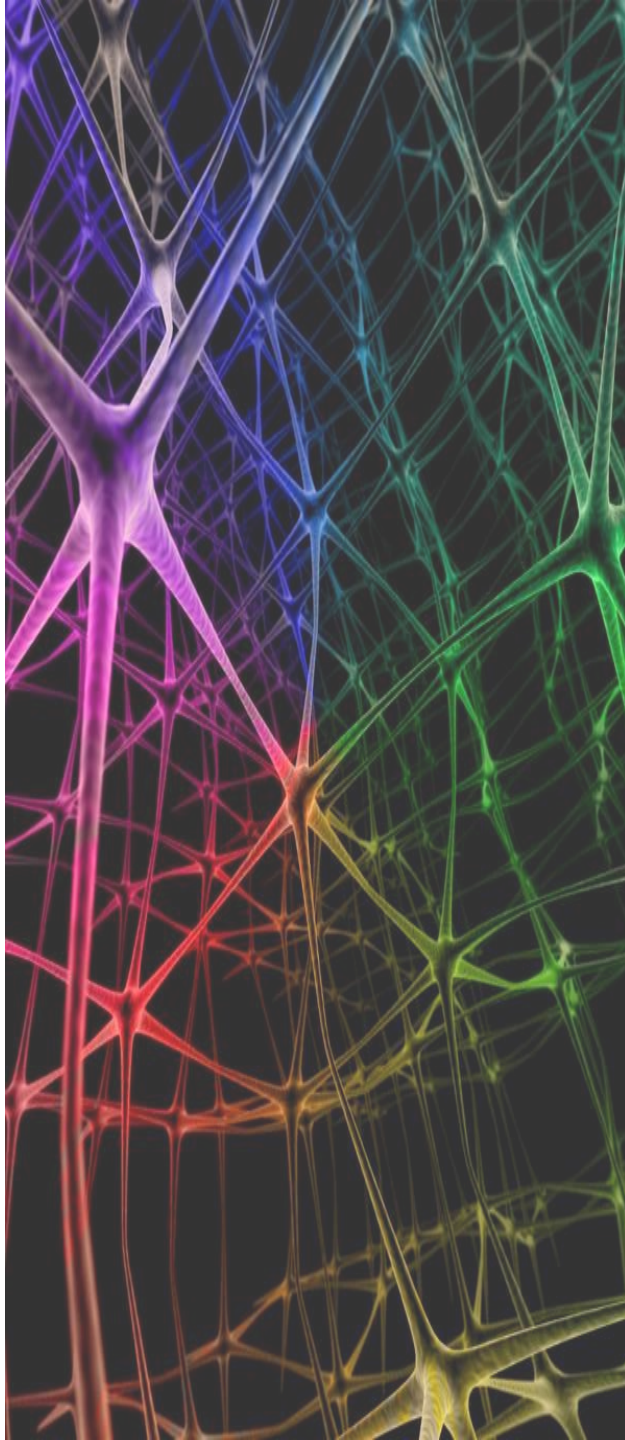


# Machine Learning – How it works



# Questions?

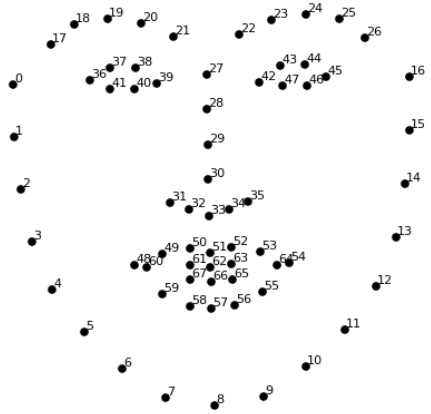
About us? Barco? Machine Learning?



# Hands-on – Introduction

Some Machine Learning, some coding and … some fun!

- Machine learning to detect facial landmarks
- Machine learning to detect face configuration and emotion
- Some code to overlay sprites on your face via your webcam



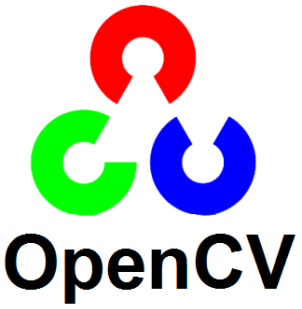
Open Mouth

# Hands-on – Quick Demo

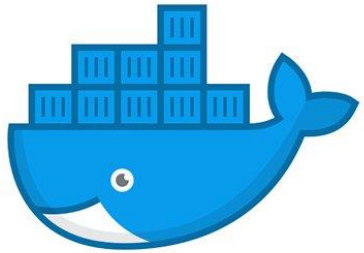
It is show time!

# Hands-on – Introduction

What will we use for this?

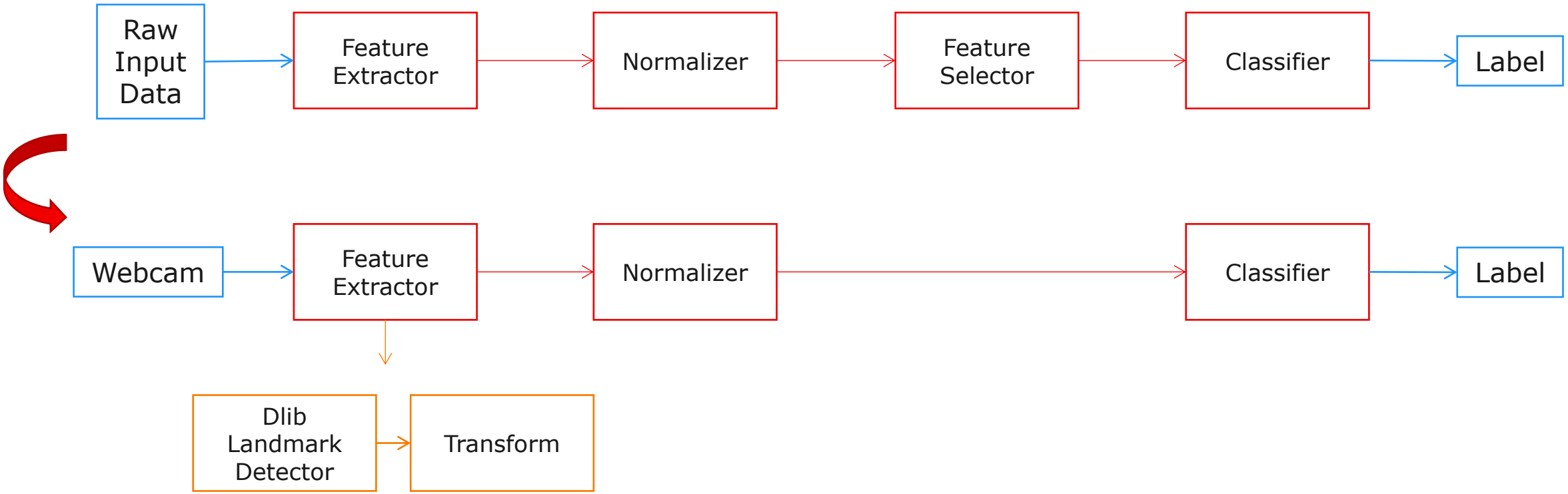



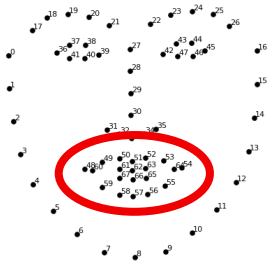


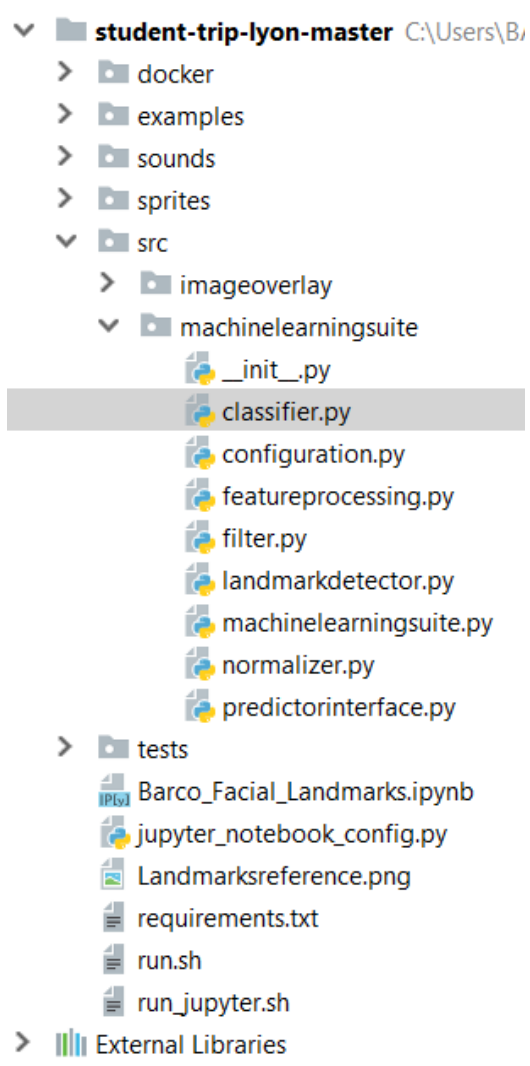
# Hands-on – Introduction

Same pipeline as before



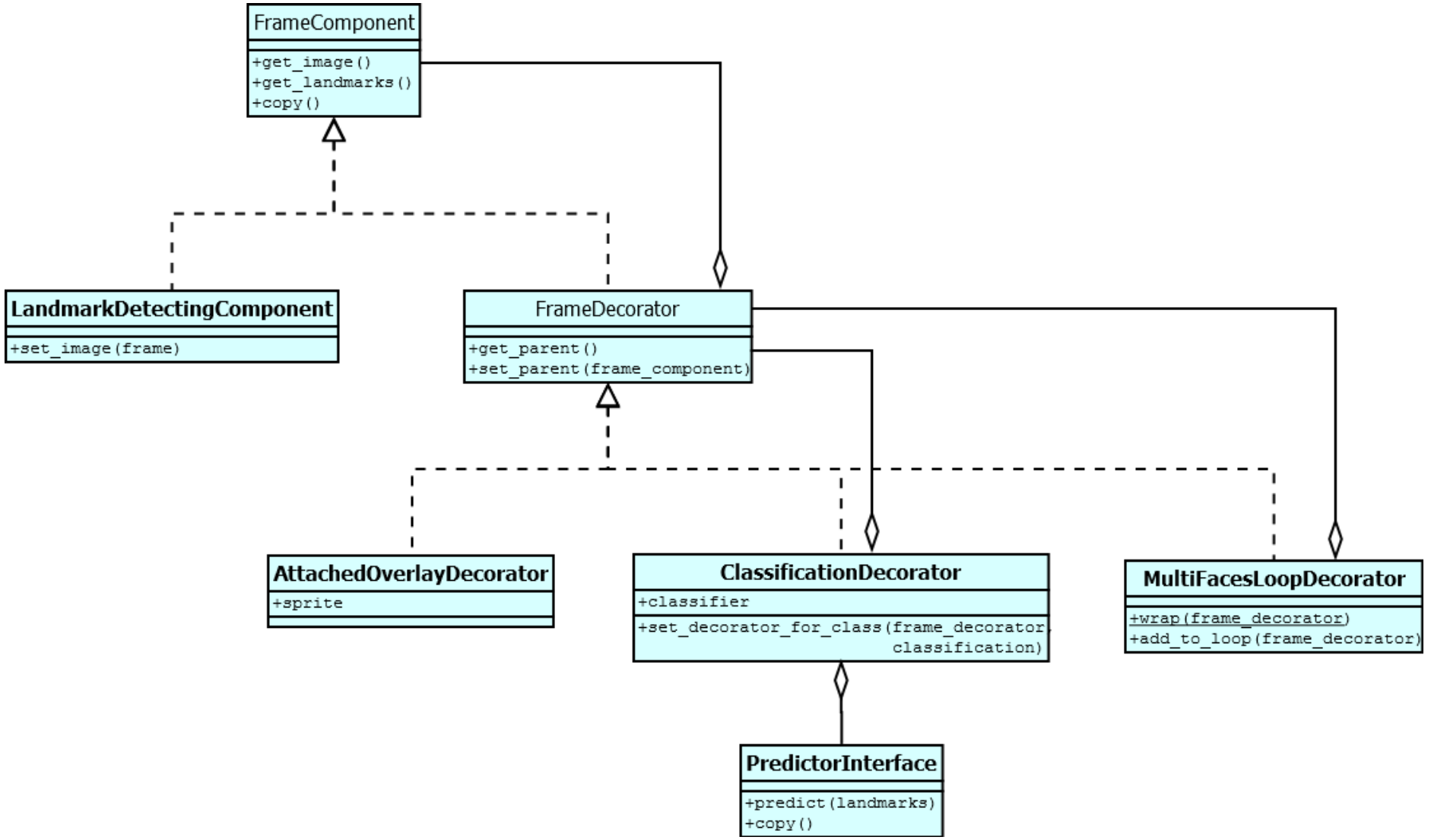



# Hands-on – Material

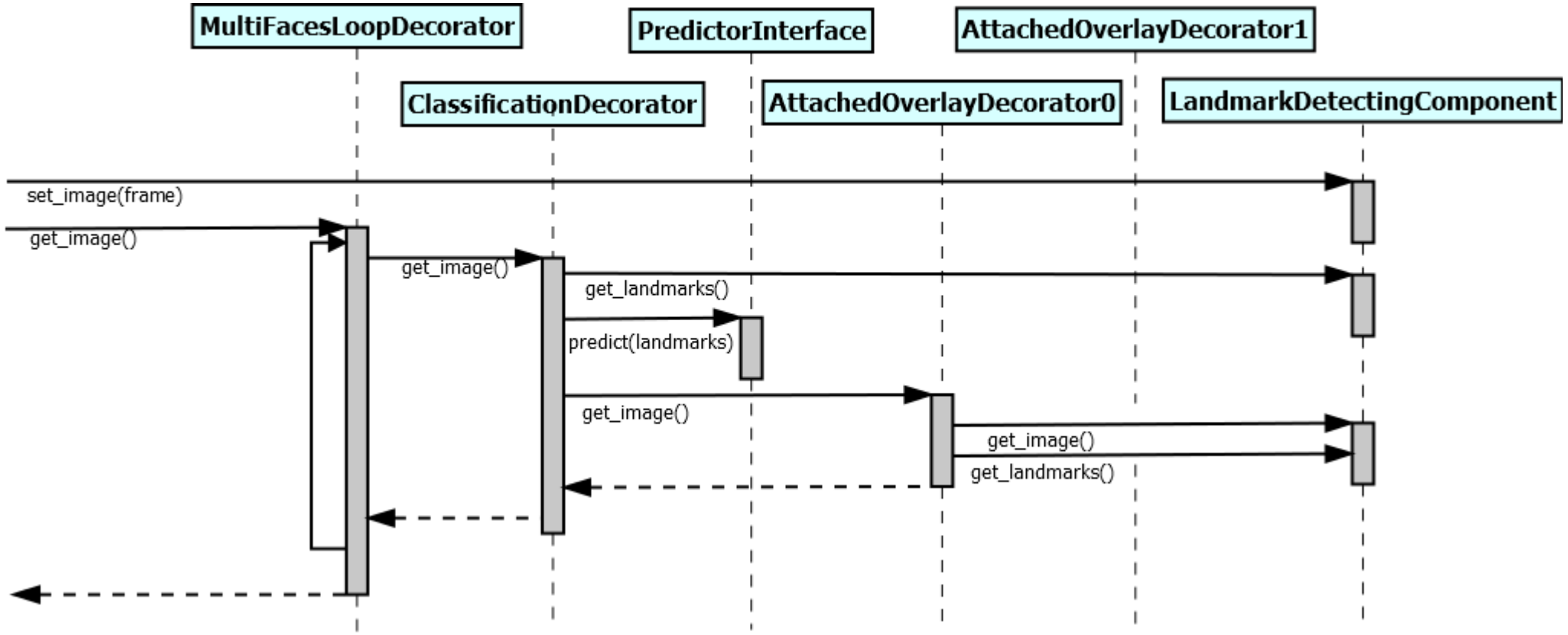



```python
class Classifier:
    def __init__(self, configuration):
        self.configuration = configuration
        self.classifier = svm.SVC(kernel='linear', C=1000)
        # self.classifier = svm.SVC()
        # self.classifier = svm.LinearSVC()
        # self.classifier = svm.SVC(decision_function_shape='ovo')

    def load_configuration(self):
        if self.configuration.classifier:
            self.classifier = self.configuration.classifier

    def save_configuration(self):
        self.configuration.classifier = self.classifier

    def train(self):
        X = self.configuration.data_values_normalized
        y = self.configuration.data_labels
        self.classifier.fit(X, y)
        self.save_configuration()

    def predict(self, data):
        return self.classifier.predict(data)
```

# Hands-on – Material



# Hands-on – Material



# Hands-on – Material

```
13      def predicting_example():
14          # Instanciate a new landmark detector
15          detector_data_path = '../../data/shape_predictor_68_face_landmarks.dat'
16          landmark_detector = LandmarkDetector(predictor_file=detector_data_path)
17
18          # Create a frame component with landmarks
19          base_component = LandmarkComponent(landmark_detector)
20
21          # Instantiate and initialize the trained predictor
22          predictor = PredictorInterface('../examples/emotion.pkl')
23          predictor.initialize()
24
25          # Add decorator for the predictor
26          predictor_decorator = ClassDecorator(parent_component=base_component, classifier=predictor)
27          sunglasses = SpriteDecorator(base_file_name='../sprites/sunglasses')
28          eyes = SpriteDecorator(base_file_name='../sprites/eyes')
29          predictor_decorator.set_decorator_for_class(sunglasses, 0)
30          predictor_decorator.set_decorator_for_class(eyes, 1)
31
32          multifaces = AllFaces.wrap(predictor_decorator)
```

```
43              base_component.set_image(frame)
44              output = multifaces.get_image()
```

# Hands-on – Get Started

- Get docker image from:
  docker pull bapha/student-trip-lyon

- Get source code from:
  git clone https://github.com/karmomoens/ml-in-cv.git

- ./run.sh

- Have a look in ./examples

# Hands-on – Practice

1. Show facial landmarks
2. Add overlay
3. Train classifier: Open/Closed mouth
4. Link overlay to classifier
5. Go crazy!

# Hands-on – Further Exercises

- Create a text overlay
- Face swap
- Draw a new sprite at run-time
- Create animations
- Program a (competitive) game (e.g. eating dots of the screen)

- Train a classifier for tilting your head in one or the other direction
- Create a nodding yes or no classifier
- Use the color of pixels to make a prediction (e.g. eye or hair color)
- Check out examples in dlib to label and train new object detectors and shape predictors

# Final Note

Thanks for attending! Hope you enjoyed it!

## ENABLING BRIGHT OUTCOMES

| youtube.com/BarcoTV

| linkedin.com/company/Barco

| twitter.com/Barco

| facebook.com/Barco