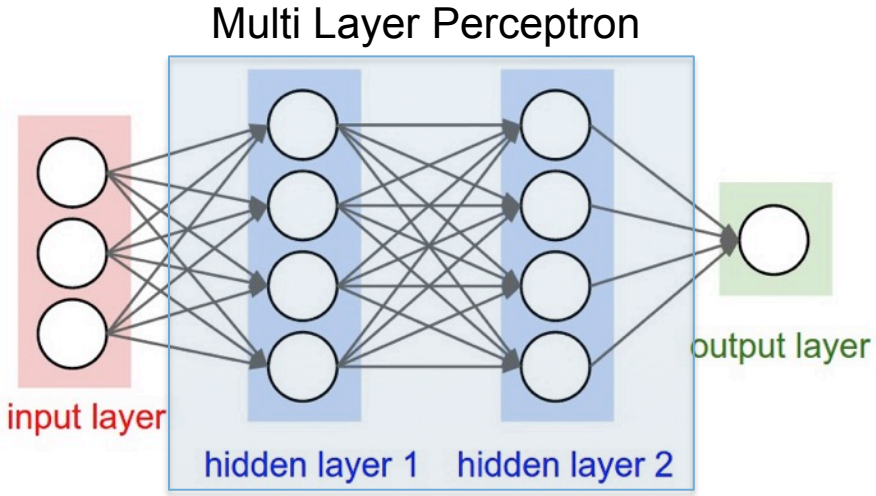
Deep Learning for the Industrial Internet

–

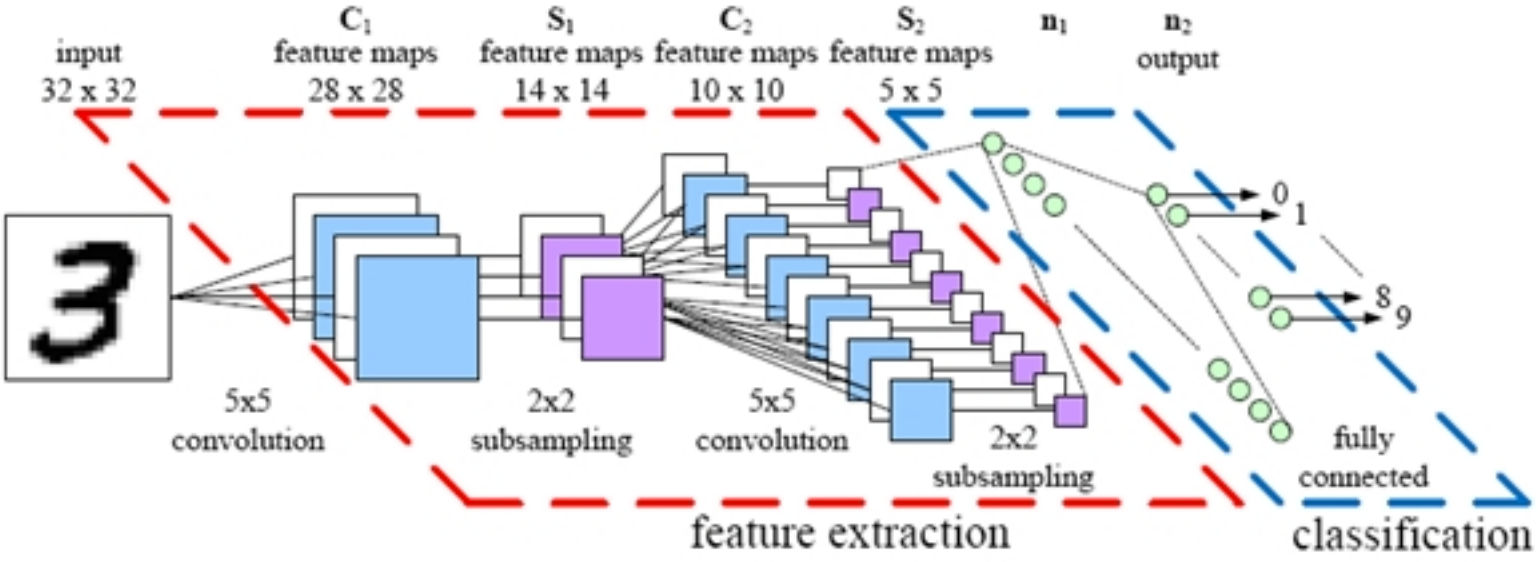
a cross-industry use case

Neural networks



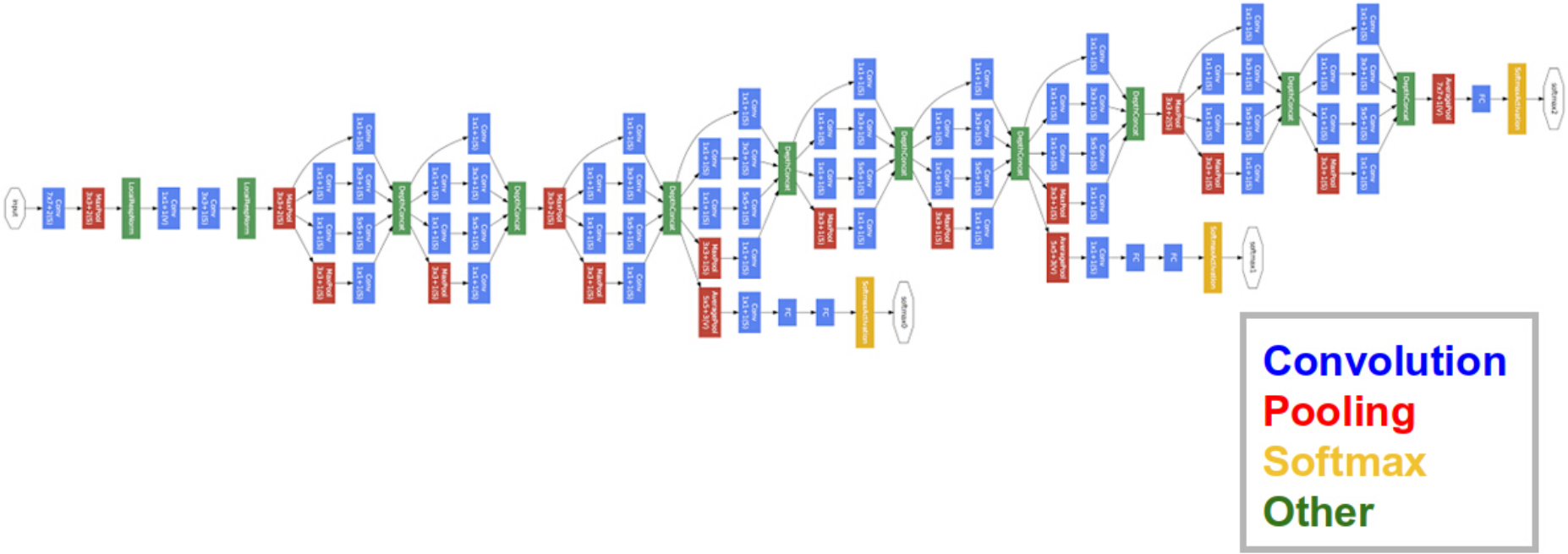
Complex mathematical functions with up to 100 Million free parameters

CNNs for image classification – handwritten digits

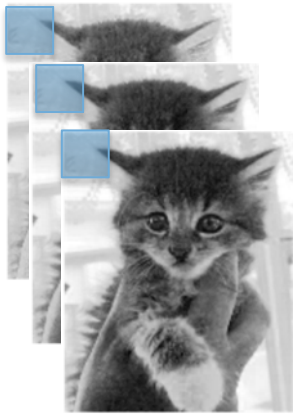


Convolutional and pooling layers

Another example - GoogLeNet



Convolution operation is actually a sliding window



$W_{5,5}$	$W_{5,4}$	$W_{5,3}$	$W_{5,2}$	$W_{5,1}$
$W_{4,5}$	$W_{4,4}$	$W_{4,3}$	$W_{4,2}$	$W_{4,1}$
$W_{3,5}$	$W_{3,4}$	$W_{3,3}$	$W_{3,2}$	$W_{3,1}$
$W_{2,5}$	$W_{2,4}$	$W_{2,3}$	$W_{2,2}$	$W_{2,1}$
$W_{1,5}$	$W_{1,4}$	$W_{1,3}$	$W_{1,2}$	$W_{1,1}$

3 x 500 x 403

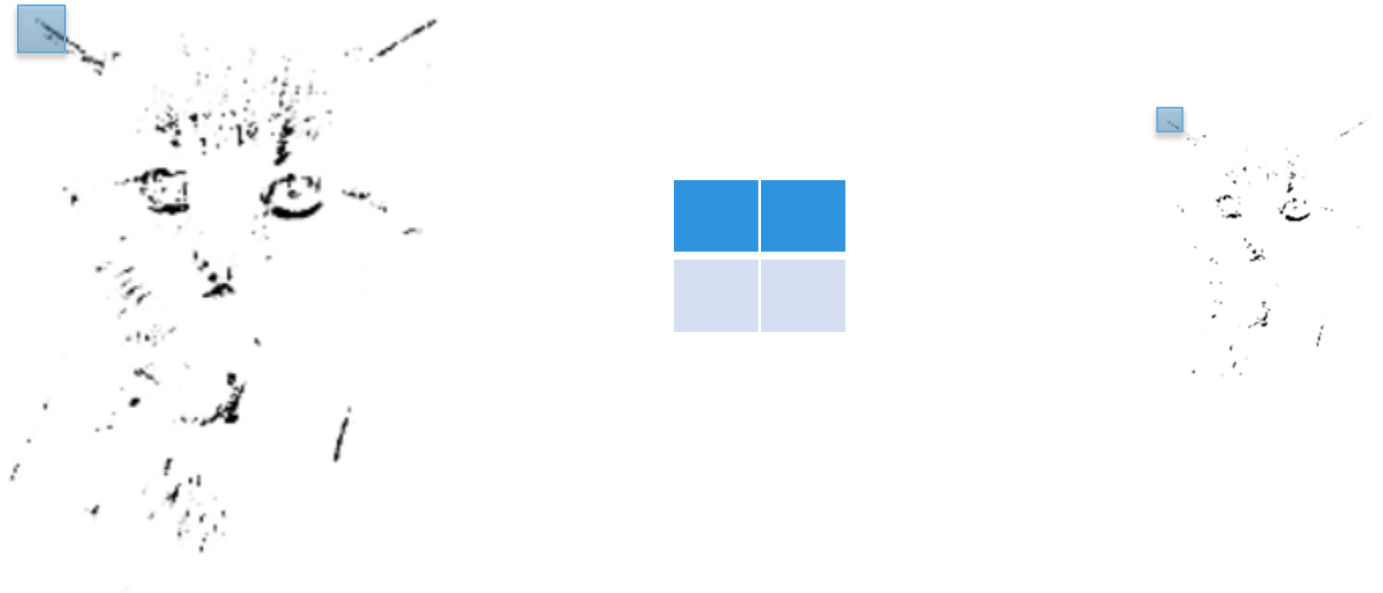
3 x 5 x 5

1 x 496 x 399

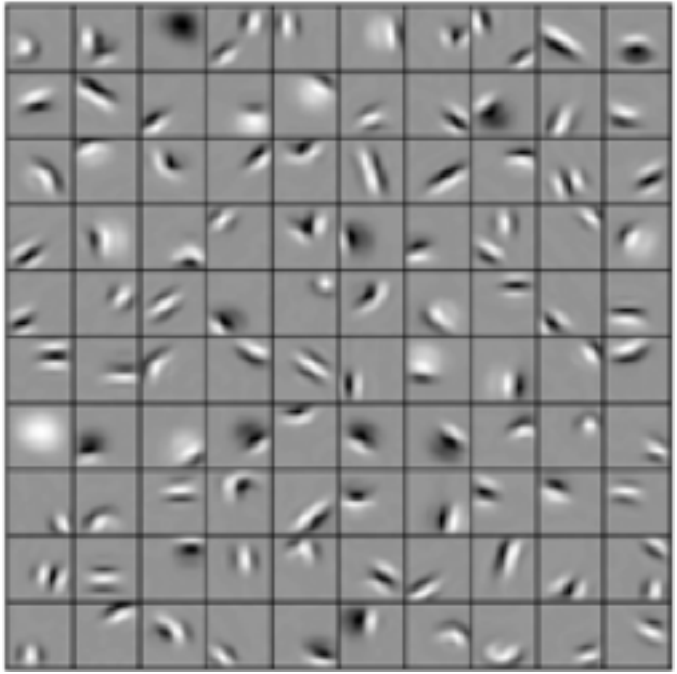
$$\sum_{k,l=1}^{5}\sum_{i=1}^{3} p^{(i)}(x+k-1, y+l-1)\cdot w^{(i)}_{5-k,5-l}$$

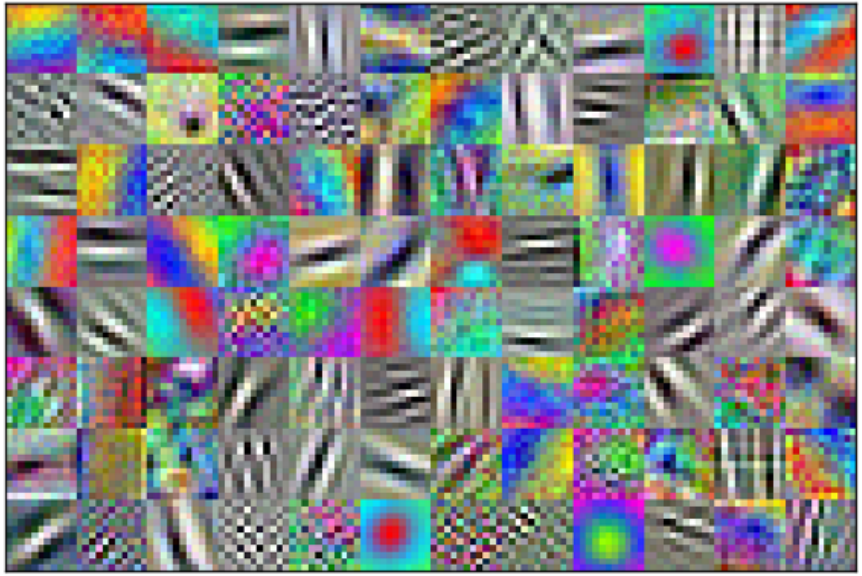
Layer 2: Maps of detected patterns

Max Pooling - subsampling layer

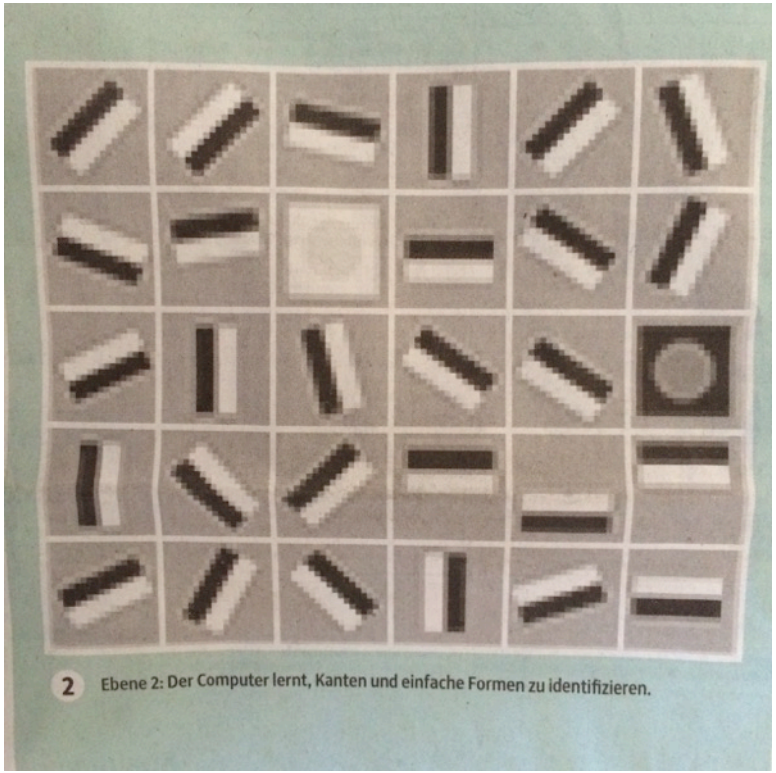


First filter layer filters





… in the newspaper



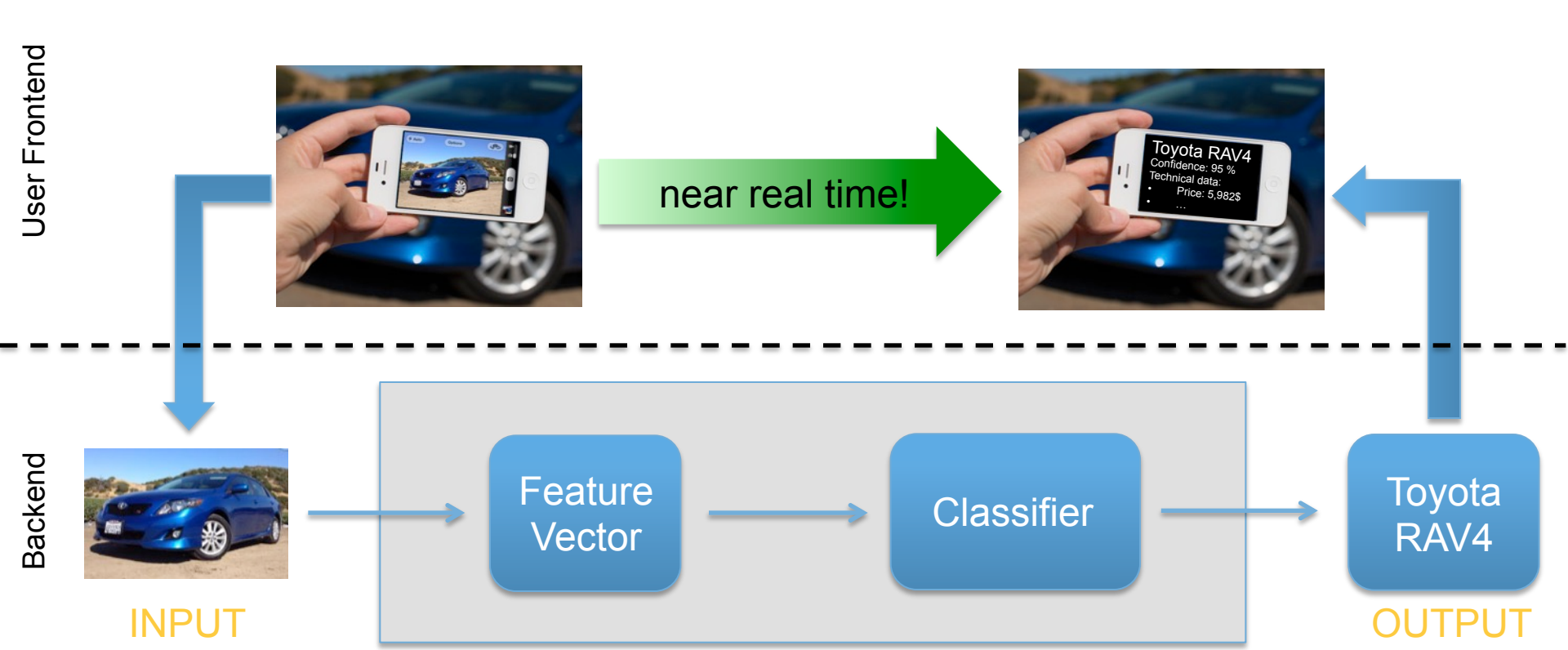
2 Ebene 2: Der Computer lernt, Kanten und einfache Formen zu identifizieren.

Our use case...

Visual Product Recognition

Problem: Field service employees want to identify a wide variety of different products and receive detailed real time information using a mobile app.

Solution: We use deep learning algorithms to identify products in a photograph taken by the user with a mobile app and return all available data to the user.



Main tasks:

1. Extract the feature vector from a given image using neural networks
2. Train a classifier using test and training data sets

What our project manager thinks we do…

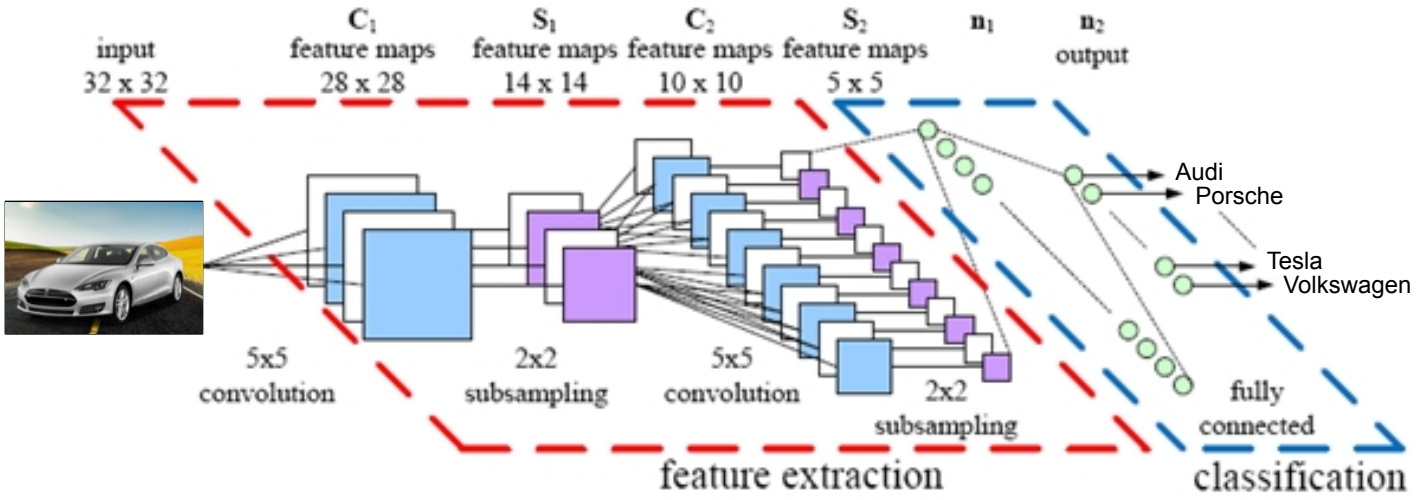
```
from sklearn import SVM
```

What we actually did…

```
from sklearn_theano.feature_extraction import OverfeatTransformer
```

Feature Vector Extraction

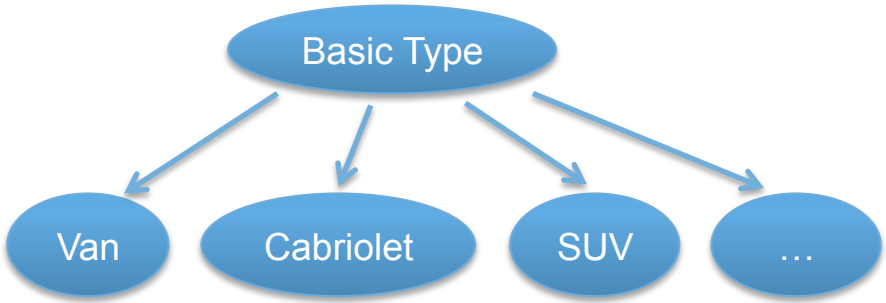
Use Google's convolutional deep neural network.



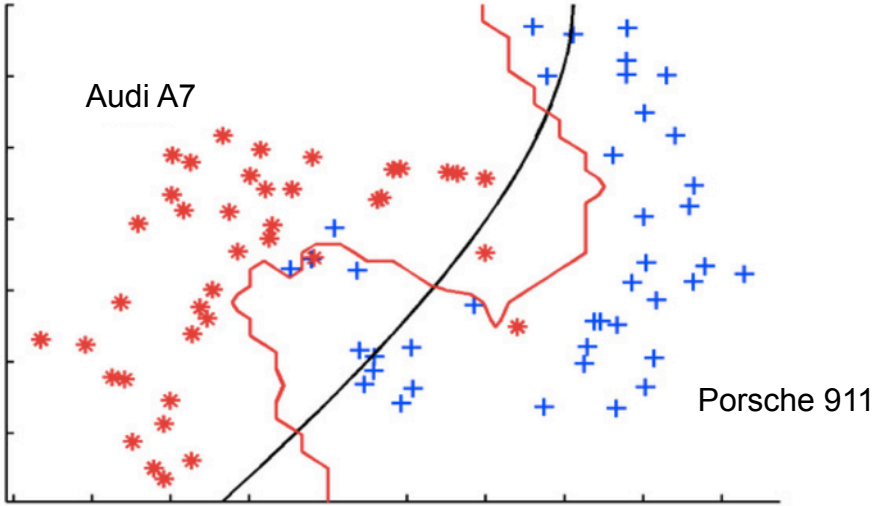
- Use of a pre-trained neural network to extract image features
- The fully connected layers are replaced by a classifier: SVM, Logistic Regression, K-Nearest Neighbor, MLP

Pre-classification

Step 1: Determine the top level category



Step 2: Determine the object



Synthetic Images

>10k images = Large Data!

Original / Translation

Reflection

Sharpness

Brightness

Rotation

Color

Perspective

Background

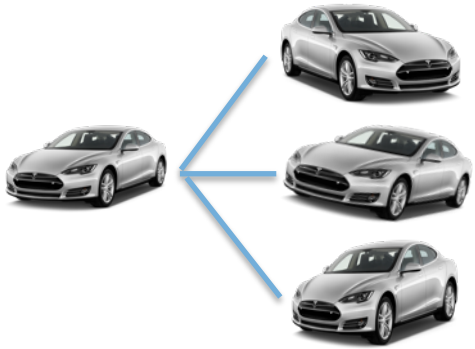
Model Tuning

Optimizing our classification accuracy

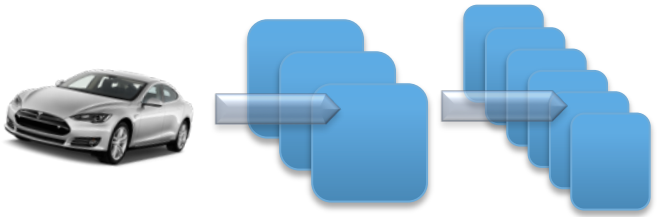
38%

optimization

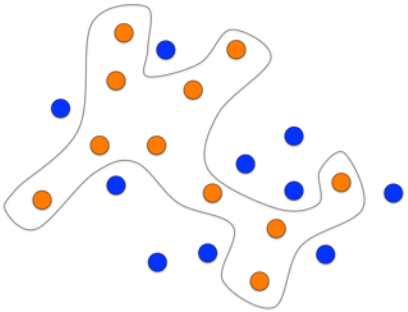
92%



Tune synthetic images
(vary parameters)

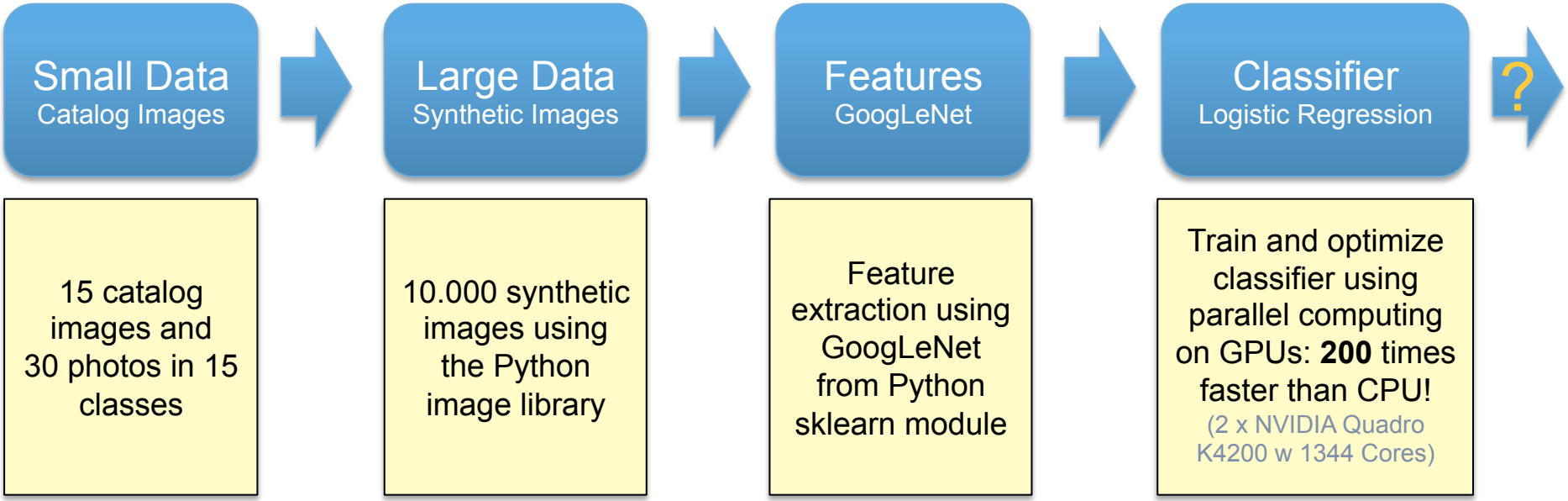


Tune image data sets
(vary feature extraction)



Tune classifier model
(vary models and structure)

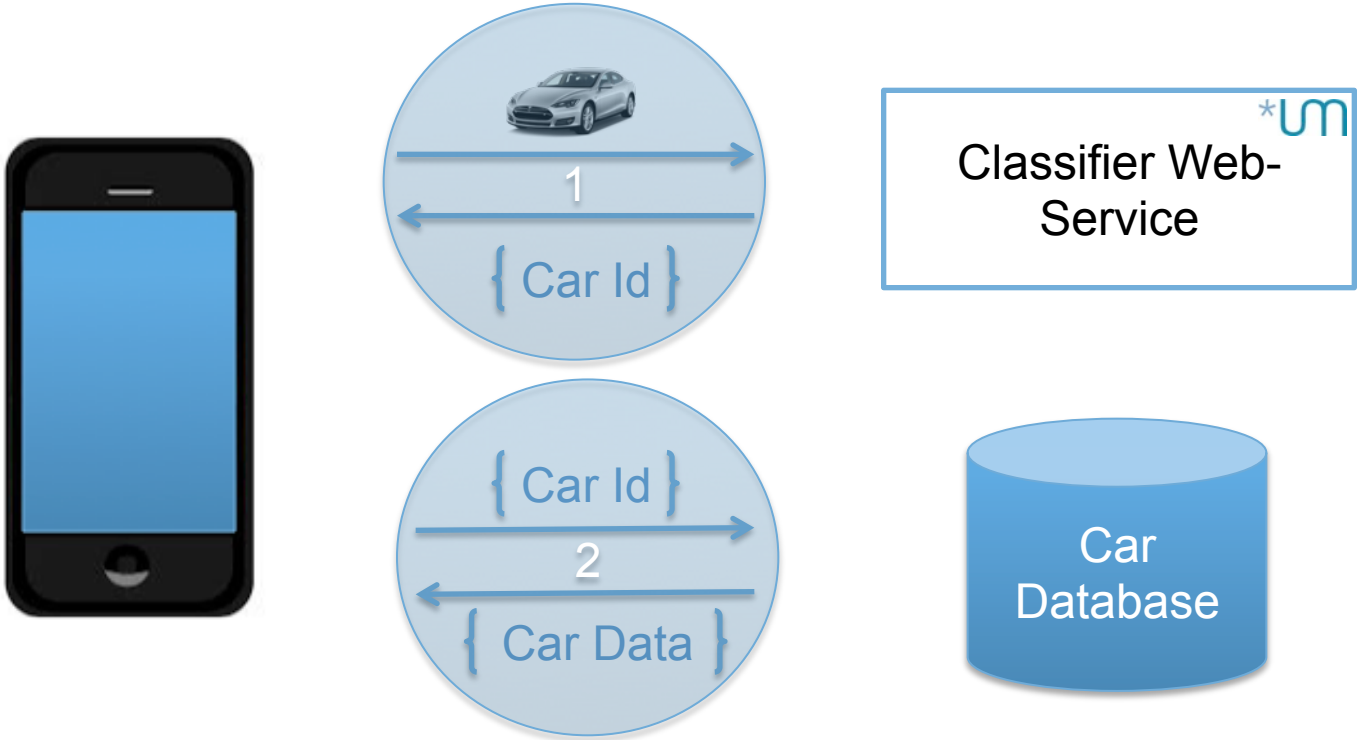
Problem solved!



The Interface matters!

Deep Learning unleashed

Connecting to external services via REST Api.



Deep Learning as a Service

With cross-industry applications.

```
from sklearn.cross_validation import StratifiedShuffleSplit
from sklearn.cross_validation import cross_val_score

cv = StratifiedShuffleSplit(y_train, test_size=0.20, n_iter=3)

%time svm_cv_scores = cross_val_score(svm, X_train_pca, y_train, scoring='f1', n_jobs=2)
svm_cv_scores
```