COMPUTER VISION 2018 - 2019
>PERFORMANCE MEASURES

UTRECHT UNIVERSITY

RONALD POPPE & ALEXANDROS STERGIOU

OUTLINE

Recap

Reporting performance

Overfitting vs underfitting

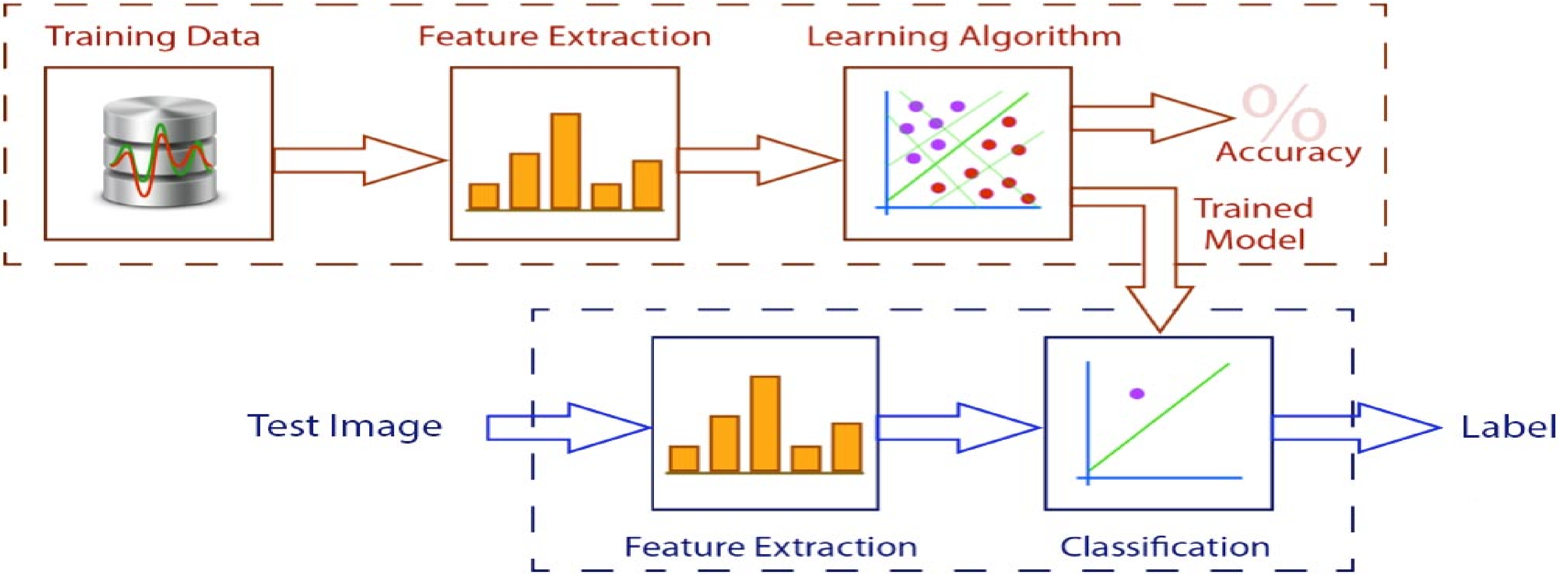
Data augmentation

Negative hard mining

Assignment

RECAP

RECAP



Training Data
Feature Extraction
Learning Algorithm
%
Accuracy
Trained Model
Test Image
Feature Extraction
Classification
Label

RECAP^2

If we are to classify images, we need a trained classifier

We can train a classifier using training data

- Supervised learning requires a dataset of pairs of image features with image labels (**x**, y)

There are many different classifiers

RECAP³

Once we have a trained classifier, we can classify images of which we do not know the label (“unseen data”)

First, we calculate image features:

- HOG
- SIFT
- Color histogram
- Etc.

Then we test/evaluate the trained classifier

PERFORMANCE MEASURES

PERFORMANCE MEASURES

Often, we want to know how good a trained classifier is

- Requires objective and insightful measures
- Always a summarization of the data
- Single measure usually doesn’t tell the whole story

We discuss performance measures

- For image classification
- For object detection

PERFORMANCE MEASURES[2]

Performance measures typically calculated on the test set

- Can also be used during validation to select the best parameters
- Can also be used during training to guide the optimization (loss function discussed next lecture)

PERFORMANCE MEASURES[3]

Consider a binary classification problem

- True class (ground truth) is either the target class or “other”
- Guess (classification outcome) is either the target class or “other”

When we calculate the performance over an entire test set

- The guess for each image is either correct or incorrect
- Accuracy: percentage of correct classifications across the test set
- Naturally between 0% (no correct classifications) and 100% (all correct)

PERFORMANCE MEASURES[6]

A class→other or other→class mistake can have a different importance

- E.g. guessing that someone is not ill whereas the person is, can have dramatic consequences

Especially when there is a skewed distribution, it is advisable to use more informative performance measures

- Allows us to put more emphasis on a minority class

PERFORMANCE MEASURES[7]

Based on the fact that an image has an actual label and a guessed label, we define:

- True positive: actual class guessed right
- True negative: other class guessed right
- False negative: actual class guessed wrong (missed detection)
- False positive: other class guessed wrong (insertion)

		Guessed	
		True	**False**
Actual	**True**	True positive	False negative
	False	False positive	True negative

PERFORMANCE MEASURES[8]

Usually, we use the precision (P) and recall (R) measures:

		Guessed	
		True	**False**
Actual	**True**	True positive	False negative
	False	False positive	True negative

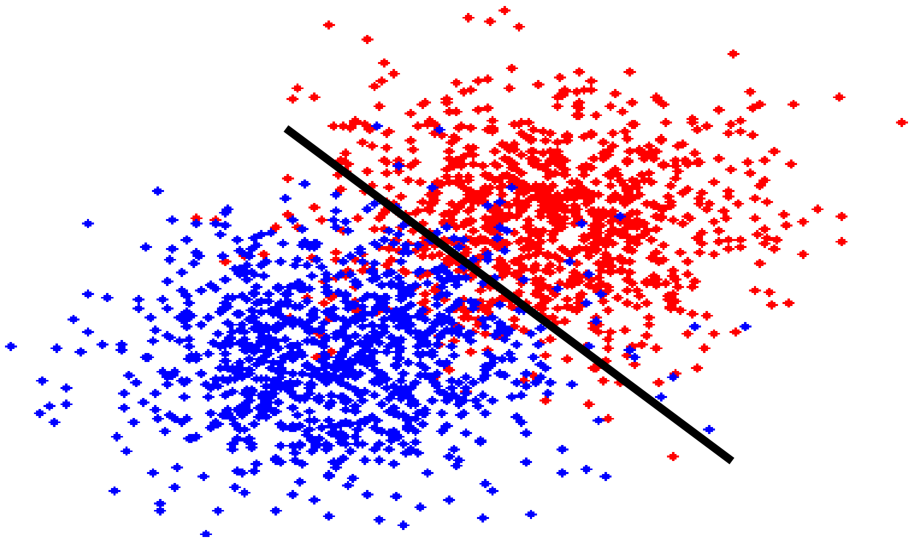
- P = TP / (TP + FP)
- Of all guesses, how many percent is correct

- R = TP / (TP + FN)
- Of all actual instances of the class, how much percent was found

PERFORMANCE MEASURES[9]

Often, there is a trade-off between precision and recall

- Due to a threshold or by changing a parameter
- As a result of varying amounts of training data

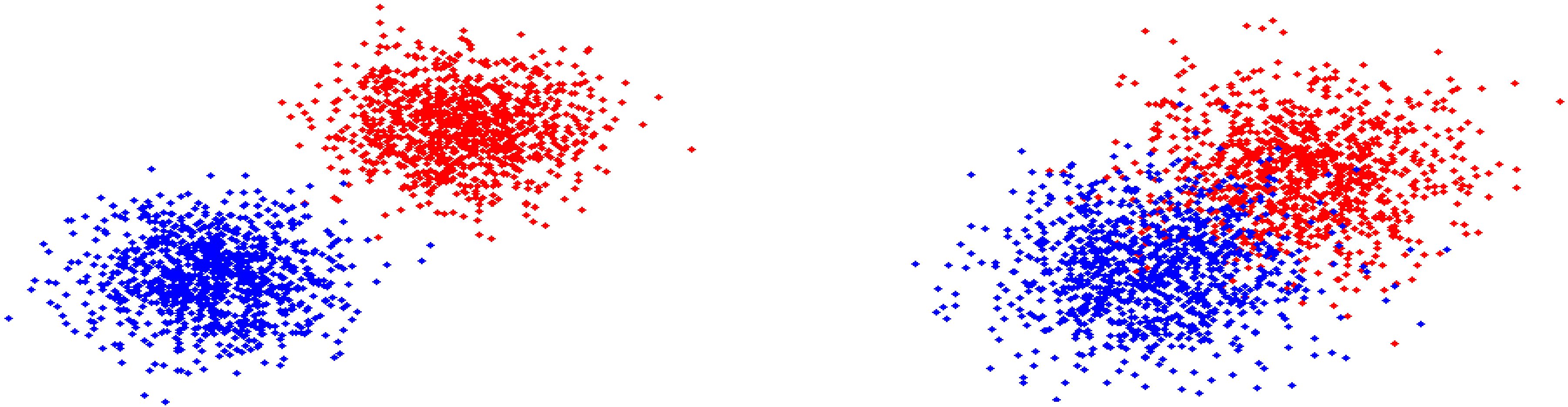


PERFORMANCE MEASURES[10]

Only 100% precision and 100% recall when classes are perfectly separable

When the other class becomes larger, precision and recall usually drop

- More difficult to identify the target class amongst the other class



PERFORMANCE MEASURES[11]

We often want to say something about both precision and recall, at the same time

- E.g. to say which of two outcomes is best, two numbers create ambiguity
- To select the best set of parameters, or to guide the training (loss function)

Three options:

- F-score
- Curve-based
- Recall@X, precision@Y

PERFORMANCE MEASURES[12]

F-score is a measure that takes both precision and recall into account

- It is sometimes called the “harmonic mean” of P and R
- Or f1-score, f-measure

F-score = 2 * P * R / (P + R)

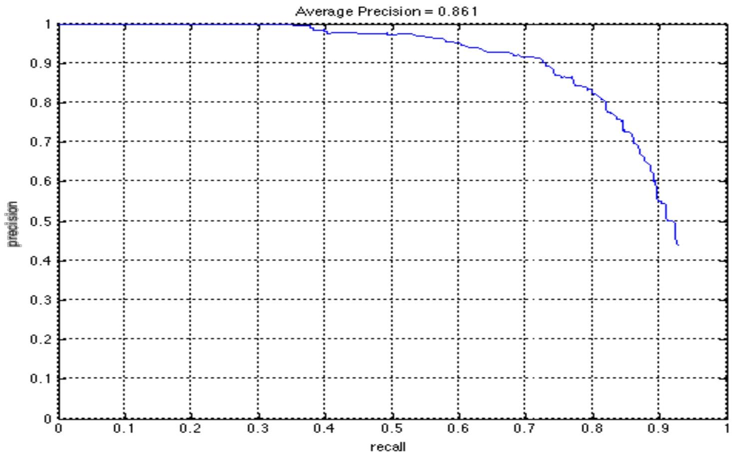
Naturally between 0 and 1

- Relatively steep decline when P or R decreases

PERFORMANCE MEASURES[13]

We can use a PR-curve to show how P and R are related as a function of the changing parameters

- E.g. amount of training data or a threshold



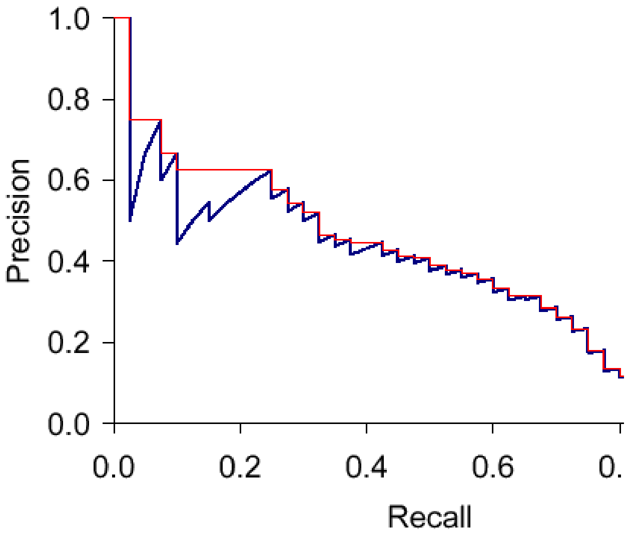
PERFORMANCE MEASURES[14]

The average precision (AP) is the area under the PR-curve

- Single number that tells us how specific our results are to a range of parameters

Some issues when calculating (AP)

- Interpolation (red vs. blue line)
- Missing values for recall 0 and/or 1



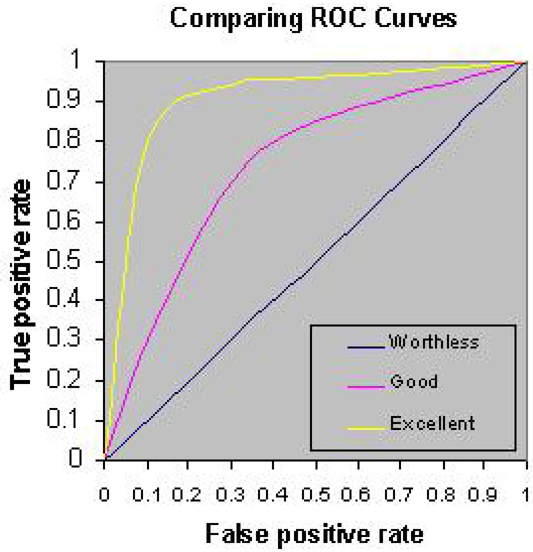
PERFORMANCE MEASURES[15]

An alternative is the receiver-operating characteristic (ROC) curve

- Y-axis: Sensitivity (recall, true positive rate) = TP / (TP + FN)
- X-axis: 1-Specificity (1 – true negative rate) = 1 – TN / (TN + FN)

Area under the curve (AUC) is single-value measure

- Calculation similar to AP

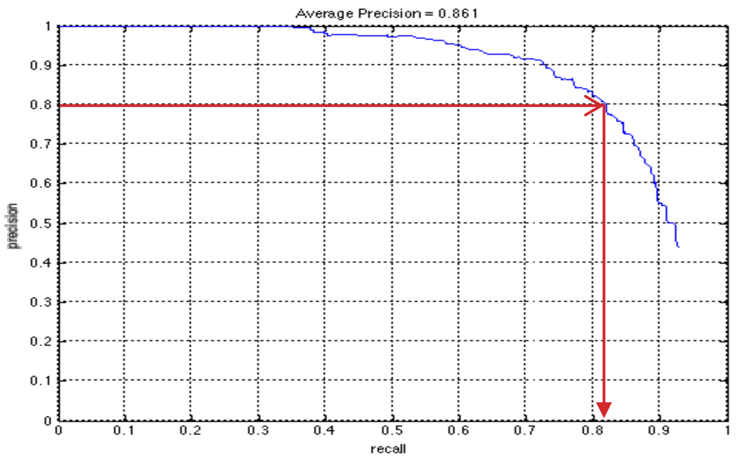


PERFORMANCE MEASURES[16]

We can also assume a specific value for either P or R and report the value on the other

Examples:

- Precision@90% recall
- Recall@80% precision



PERFORMANCE MEASURES[17]

For a multi-class problem

- Image class is from a limited set of class labels

We can still consider a guess correct or incorrect

- Not all mistakes might be equally bad
- Biases in the class distribution might go unnoticed

We can also look at the type of mistakes/confusions

PERFORMANCE MEASURES[18]

A confusion matrix shows these confusions

- Rows: true class labels
- Columns: guessed class labels

Accuracy can be calculated by dividing the sum of the diagonal by the sum of all cells

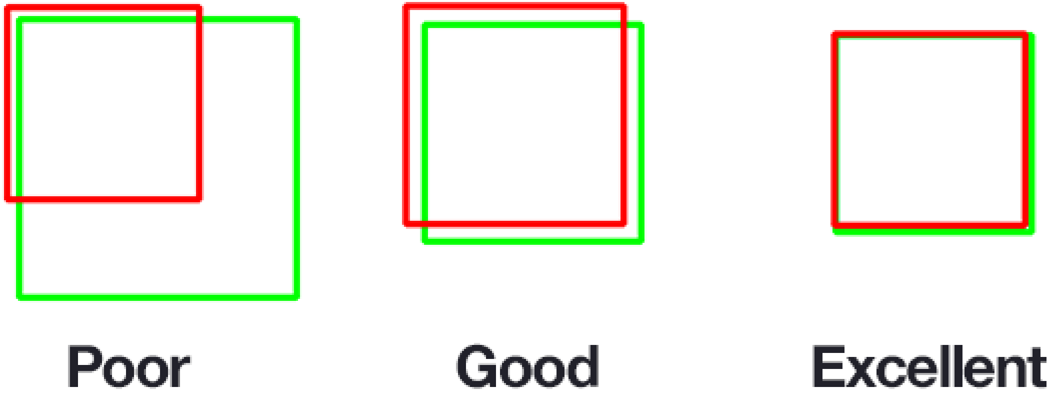
Estimated/guessed class

True class		Ferrari	McLaren	Daihatsu
	Ferrari	40	7	3
	McLaren	8	30	2
	Daihatsu	4	1	5

PERFORMANCE MEASURES[19]

For object detection, we estimate the object class and location (bounding box)

- Introduces additional complexity regarding position and size
- Requires a criterion what constitutes a “match”



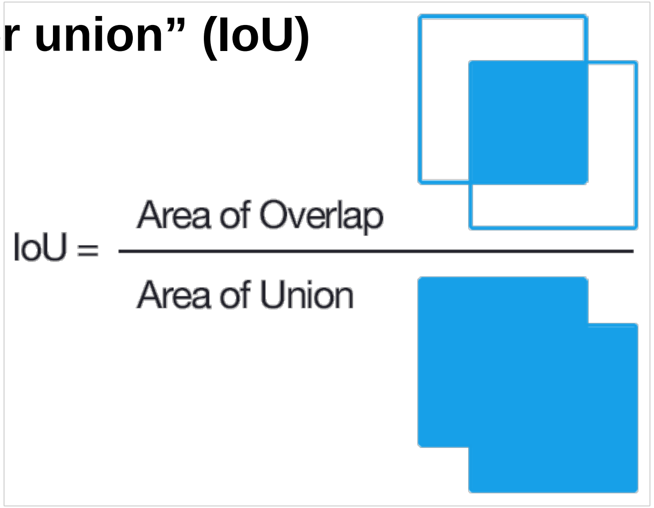
PERFORMANCE MEASURES[20]

We say that a guess is correct if it “sufficiently” overlaps with the actual (ground truth) location

- Requires a threshold

Sufficient can be percentage of “intersection over union” (IoU)

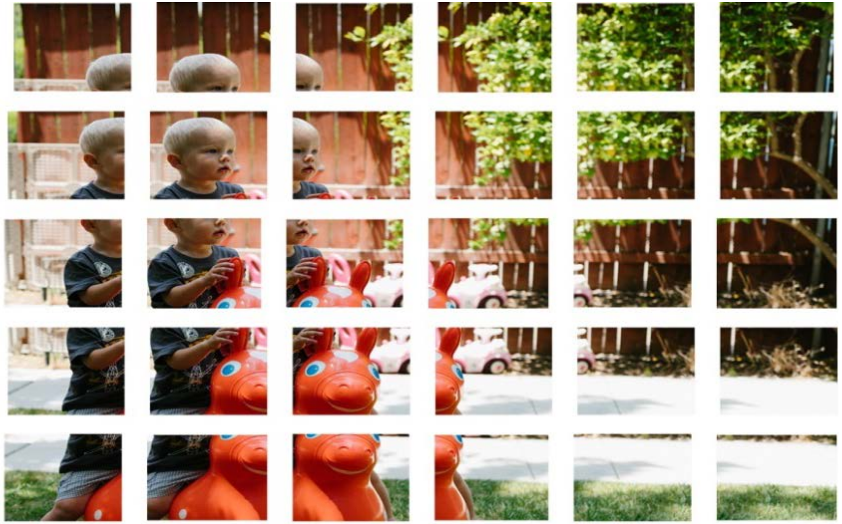
- Divide the area of overlap by the area of union
- Area of union is area1 + area2 – overlap1-2
- IoU naturally between 0 and 1



PERFORMANCE MEASURES[21]

For object detection, we typically evaluate many different regions

- Slightly different scales and positions
- Many detections can represent the same object



PERFORMANCE MEASURES[22]

We need to filter “duplicate” object guesses out

- Ideally, we end up with one guess for each object of interest

When we detect objects, the assumption is that we have a score that indicates the classifier’s confidence

- Non-maximum suppression is an algorithm that filters out duplicates based on these detection scores

PERFORMANCE MEASURES[23]

After object detection (sliding window, selective search, etc.), we have a list of detections:

- Bounding boxes with associated detection scores

Basic idea of Non-Maximum Suppression (NMS):

- Sort detections based on detection score (highest first)
- Iteratively remove bounding boxes that overlap with those with higher detection scores

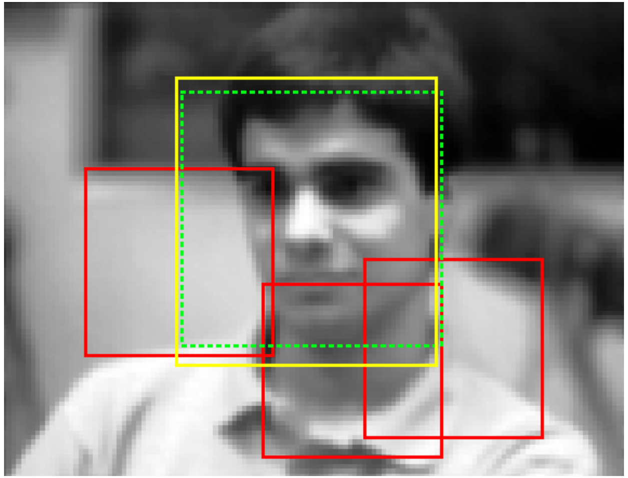
PERFORMANCE MEASURES[24]

Conceptually: keep the best detections at a certain area, across variations in position and size

Minimum overlap to remove detections determined empirically

- Intersection over union
- Typically set to 0.5

Result is a limited list of detections



PERFORMANCE MEASURES[25]

Based on the remaining detections, we can again say whether it is correct or not

- Matching problem turned into binary problem
- All previously discussed performance measures apply

Alternatively, we can decide not to filter, and use other measures:

- False positives per window (FPPW)
- False positives per image (FPPI)

Typically, curves such as missed detections vs. FPPW are used

QUESTIONS?

OVERFITTING VS UNDERFITTING

OVERFITTING VS UNDERFITTING

Ideally, our machine learning model generalizes perfectly on a validation/test set

Overfitting occurs if our trained model is tailored to the training data

- Usually too many parameters for the amount of training data

Underfitting occurs if the complexity of our model is too low

- Usually too few parameters to model the difference between classes

OVERFITTING VS UNDERFITTING[2]

Finding the right balance between overfitting and underfitting is difficult

- In practice, always found empirically

When iteratively training a machine learning model, consider performance on training and validation set

- If scores start to diverge: overfitting

Underfitting can only be identified when iteratively increasing the machine learning model's complexity and observing the scores

DATA AUGMENTATION

DATA AUGMENTATION

A training set should be representative of the application domain

- Cover relevant variations in nuisance factors

Adding more data can help to cover more variation

- Sometimes not possible

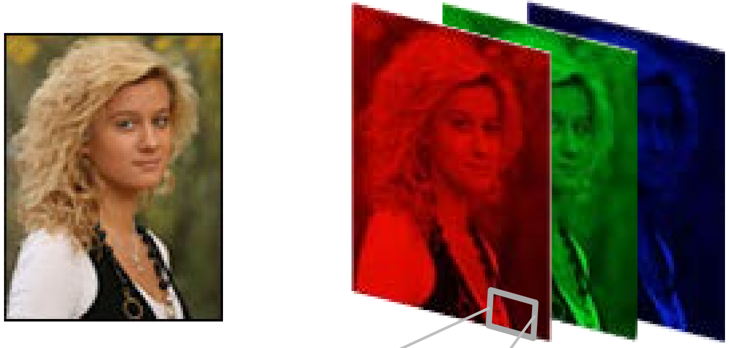
Using data augmentation, we synthetically inflate the variation

- No new images, just variations (transformations) of our original training data

DATA AUGMENTATION[2]

The primary goal for the creation of new data is:

- Changing the pixel values without changing the image label
- So label distribution remains the same but number of images per class increases
- Ideal for stratification



34	185	207	21	36
148	52	24	147	123
250	74	214	278	41
0	78	51	247	255
72	74	136	251	74

DATA AUGMENTATION[3]

Horizontal flips

- Straight-forward technique
- Doubles the size of the dataset

Make sure mirroring is justified

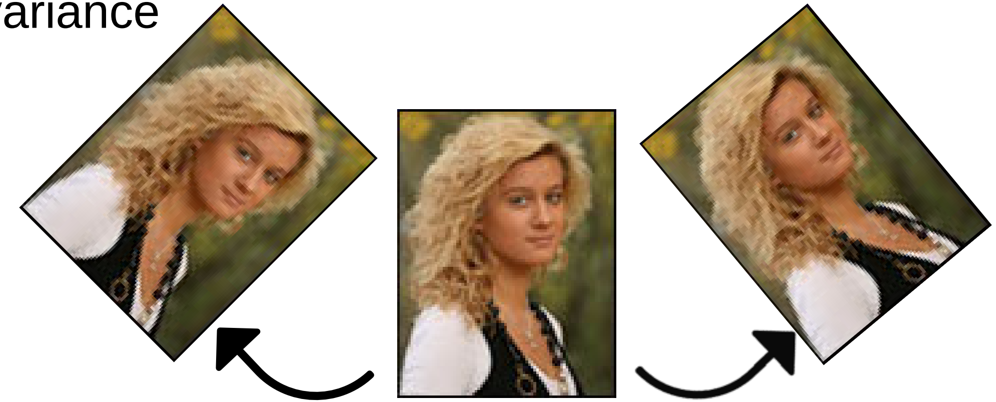
- Can we mirror an image of two people shaking hands?

DATA AUGMENTATION[4]

Angle rotation

- Defining a number that the image can be rotated/tilted (left-right)
- Typically small angles (-30 – 30)

- Effectively targets rotation invariance



DATA AUGMENTATION[5]

Random crops

- Select parts of the image by cropping and resizing it
- Make sure crops contain “enough” of the object
- Effectively targets translation invariance

DATA AUGMENTATION[6]

Color jitter

- Randomly jitter contrast to produce new images
- Can also be performed per color channel

- Can also be done locally

- Effectively targets lighting invariance

DATA AUGMENTATION[7]

There are plenty of other ways of performing data augmentation

- Stretching, shearing
- Distortions, blending of images

Most importantly:

- Different techniques can be combined
- Significantly increases the number of transformed images

Remember: there is no real substitute for additional images, but data augmentation can help

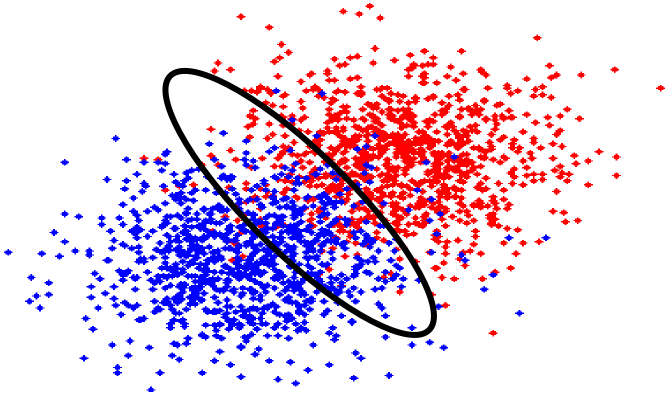
QUESTIONS?

HARD NEGATIVE MINING

HARD NEGATIVE MINING

When we evaluate a trained classifier, it is likely to return false positives

- Images that resemble (in some way) the target class
- Ideally, we learn from these mistakes!



HARD NEGATIVE MINING[2]

We can retrain the classifier using these false positives

- Ensure test data is not used for training!

Typical for training object detectors

Recipe:

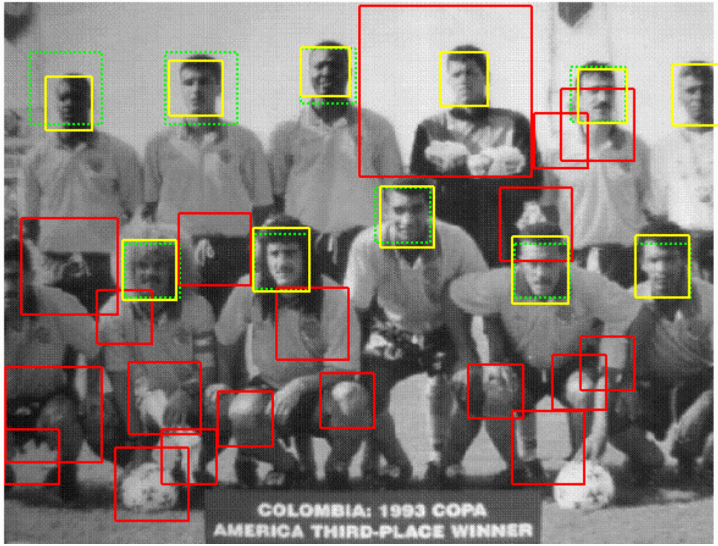
- Train object detector using initial positive and negative training images
- Run object detector on training set and find false positives
- Re-train object detector using augmented training set

HARD NEGATIVE MINING[2]

Hard negative mining can be used iteratively

Risk that number of positive samples is eventually too small

- This causes undersampling
- Overfitting can then occur



QUESTIONS?

ASSIGNMENT

ASSIGNMENT

Assignment 4:

- Essential piece of Python coding
- Will be interactively discussed in first practical session
- Tuesday March 19, 13:15-15:00, **BBG-209**
- Deadline Sunday March 24, 23:00

ASSIGNMENT^2

In Assignment 5, you will develop a pipeline to train/test CNNs

- Choice for ANN if you do not have a GPU: contact Alex Stergiou
- Coding in Python: TensorFlow and Keras
- Deadline Sunday April 14, 23:00

Two practical sessions:

- Tuesday March 19, 13:15-15:00 BBG-209
- Tuesday March 26, 13:15-15:00 BESTUURS-LIEREGG

ASSIGNMENT[3]

Action recognition: determining what a person in an image does

- Stanford 40 Action dataset: 40 actions, ~6k images

ASSIGNMENT4

In a nutshell:

- Train and test a CNN/ANN pipeline for action recognition
- Evaluate various algorithmic improvements
- Develop an algorithm for parameter search

Reporting is important:

- Motivate your choices
- Document your results (with graphs and tables)
- Reflect on your choices and results

COORDINATION

From here on, Alex Stergiou will teach lectures, practical sessions and will provide feedback for and grade the assignments

- Contact him at a.g.stergiou@uu.nl or use Slack

For all organizational matters, contact me at r.w.poppe@uu.nl

See you at the Exam Q&A lecture on Tuesday April 2

NEXT LECTURE

Next lecture:

- Thursday March 14, 11:00-12:45, RUPPERT-042
- Neural networks