Conformal automation for air traffic control using convolutional neural networks

van Rooijen, S. J.; Ellerbroek, J.; Borst, C.; van Kampen, E.

Publication date
2019

Document Version
Accepted author manuscript

Published in
Proceedings of the 13th USA/Europe Air Traffic Management Research and Development Seminar 2019, ATM 2019

Citation (APA)
van Rooijen, S. J., Ellerbroek, J., Borst, C., & van Kampen, E. (2019). Conformal automation for air traffic control using convolutional neural networks. In *Proceedings of the 13th USA/Europe Air Traffic Management Research and Development Seminar 2019, ATM 2019: 17/06/19 - 21/06/19 Vienna, Austria*

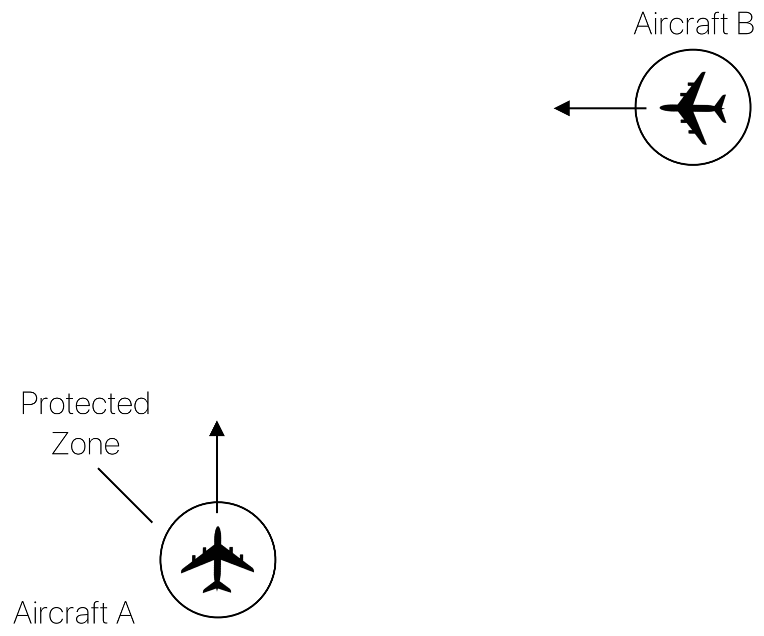
Conformal Automation for Air Traffic Control using Convolutional Neural Networks

J. Ellerbroek

TU Delft

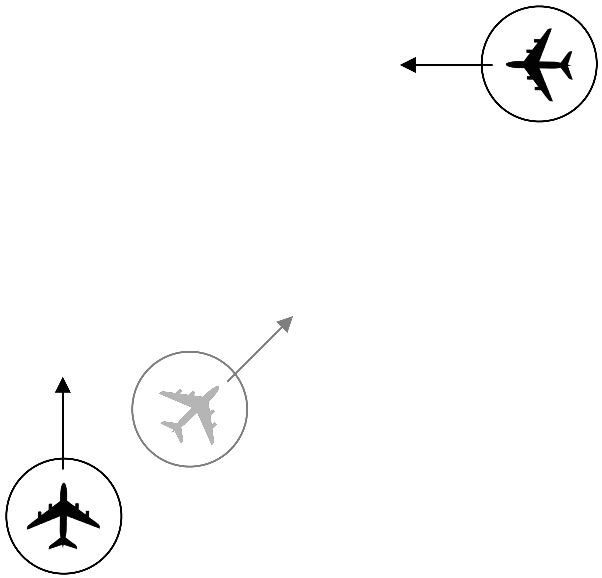
The problem of conformance

Solving a simple conflict:



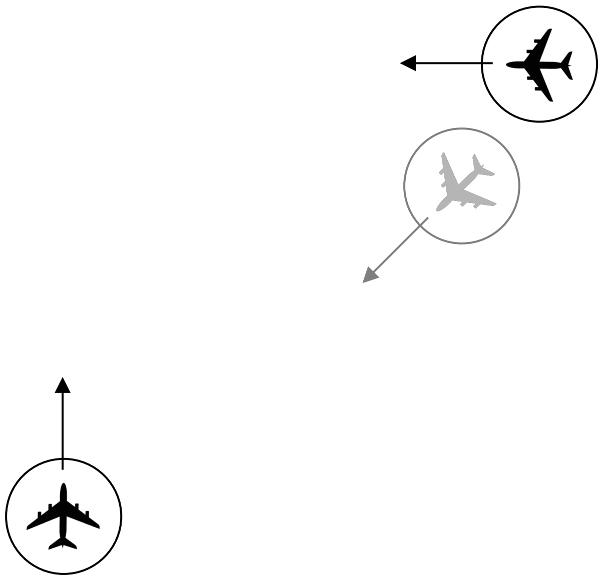
The problem of conformance

Put A behind B ...



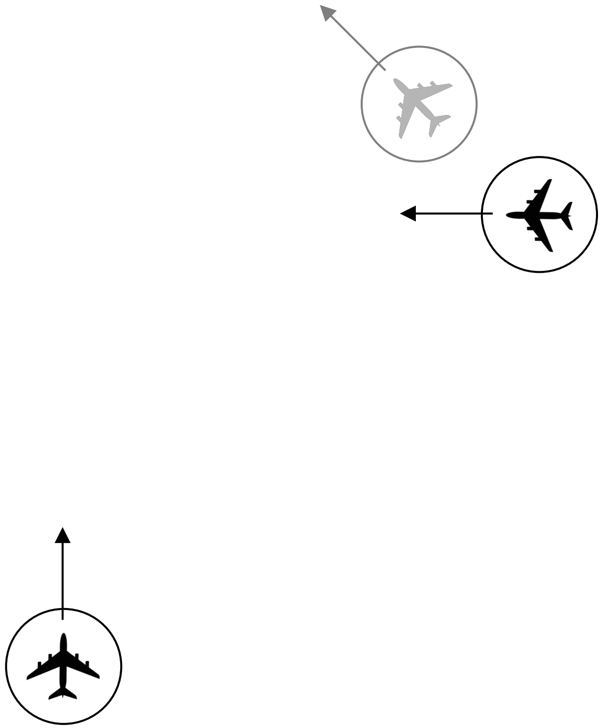
The problem of conformance

... B behind A



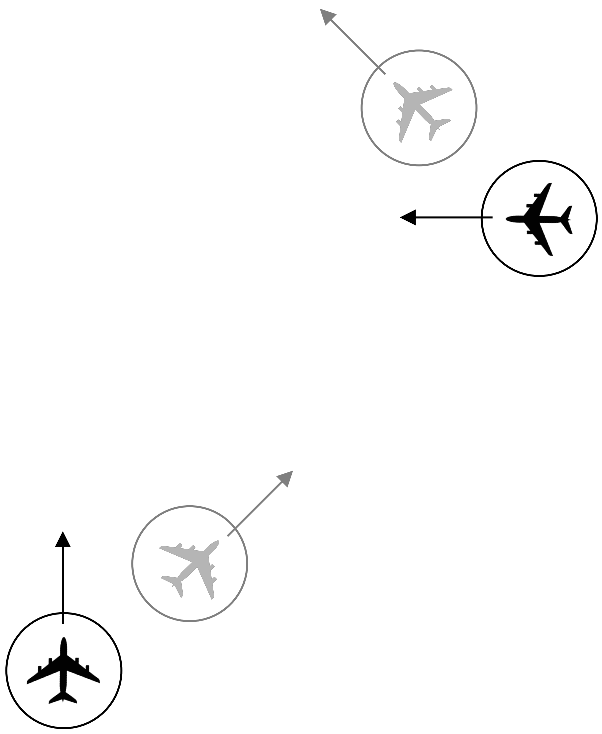
The problem of conformance

… B in front of A



The problem of conformance

... or even changing both headings

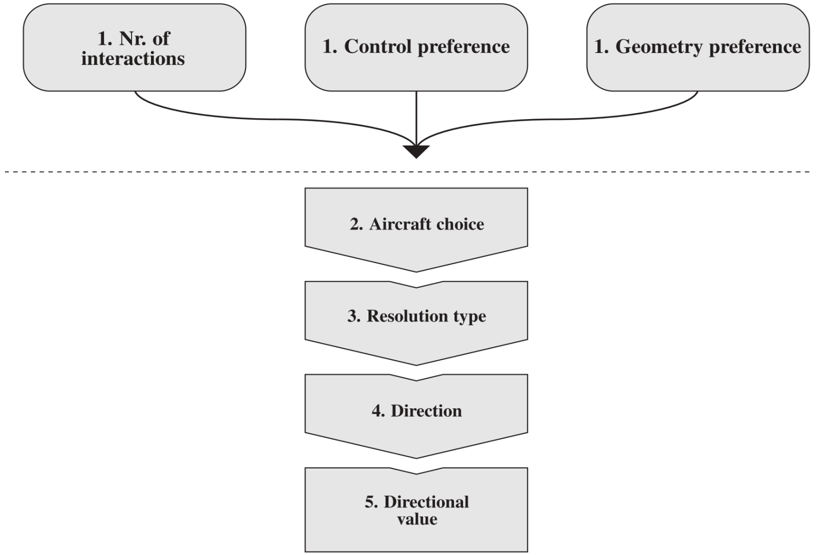


Solution: Strategically conformal automation

Strategic conformance is "... the degree to which automation's behavior and apparent underlying operations match those of the human."

– Hilburn et al. (2014)

Five levels of abstraction

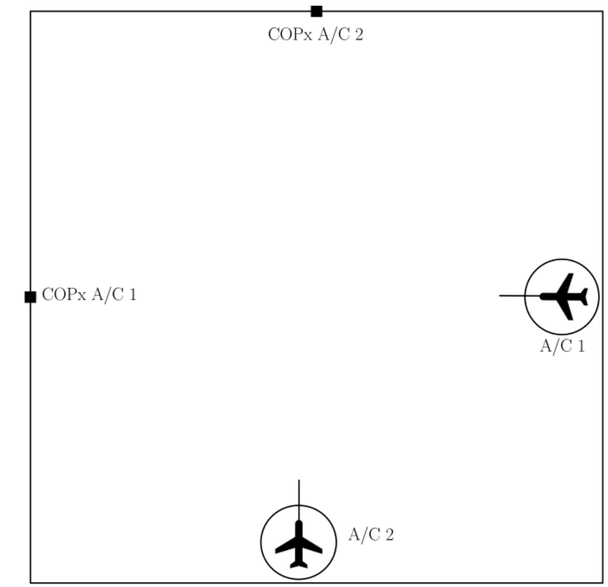


Previous research

1. Ability to adapt to controller preferences **without full knowledge** of underlying decision-making dynamics

2. Required **feature engineering** based on prior knowledge and assumptions

3. Only considered simple two-aircraft conflicts



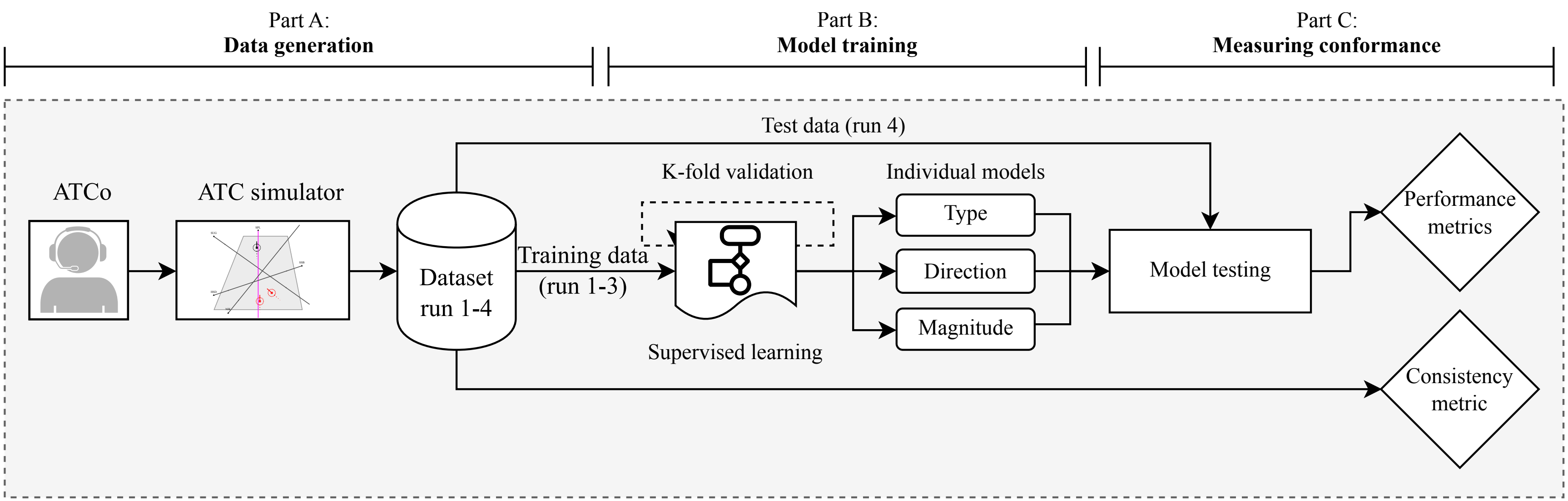
Source: Regtuit – *Strategic Conformal Automation for Air Traffic Control* (2018)

TU Delft

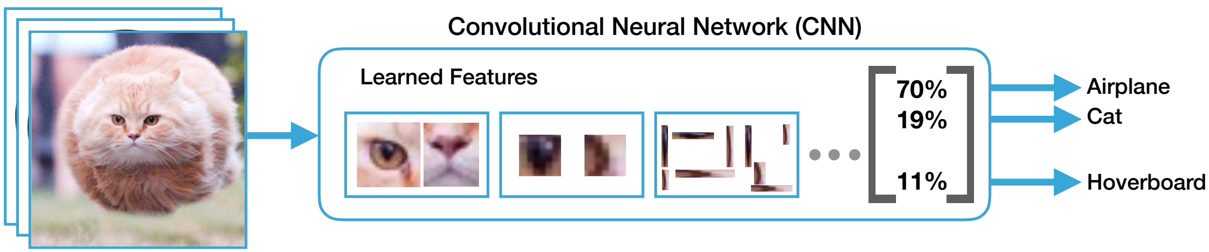
Purpose of this study: predict controller actions in a more realistic setting

Approach

Train a model on controller actions using **supervised learning**



Approach: Convolutional Neural Networks



Source: MathWorks – Deep Learning - Convolutional Neural Networks

TUDelft

Approach: Convolutional Neural Networks

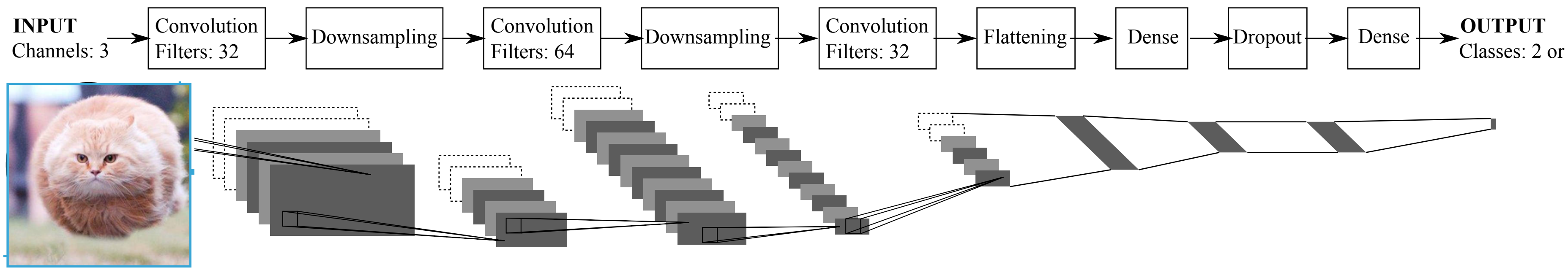


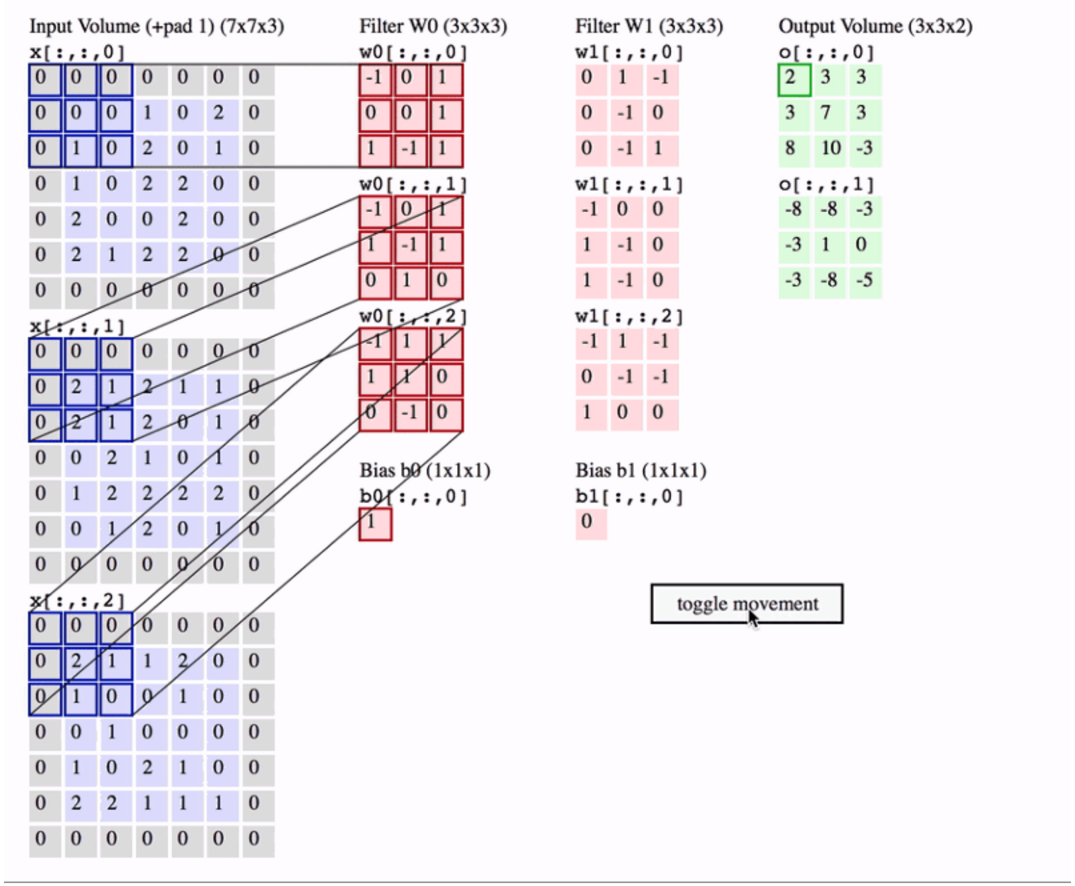
Image convolution

Filter: $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

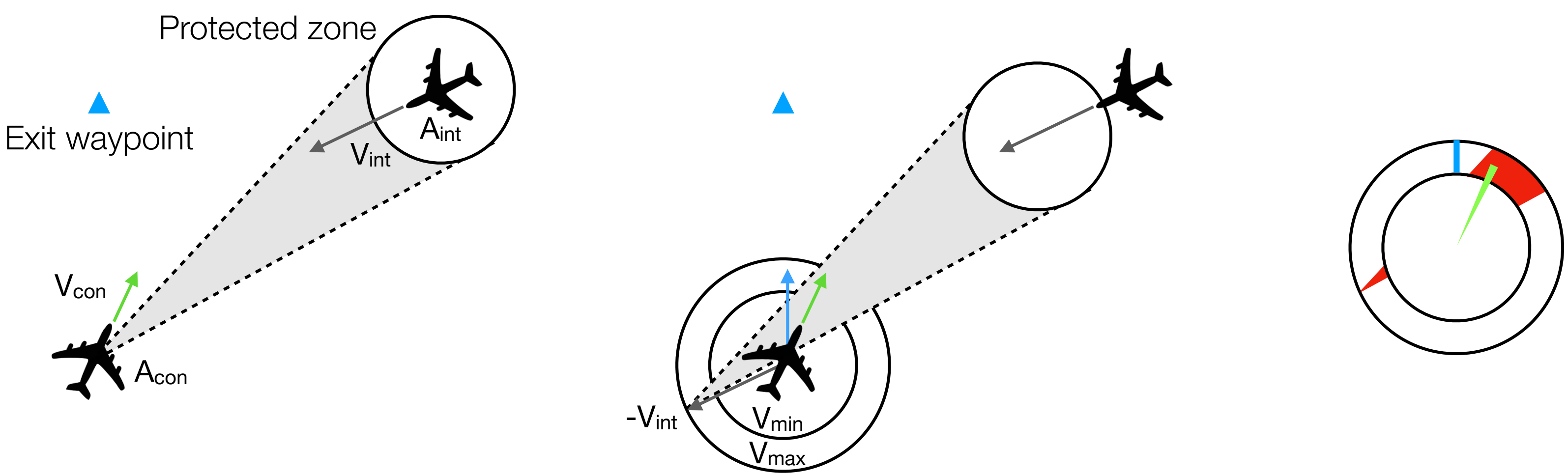
4		

A whole convolution layer



Approach: Choosing the right input

The Solution-Space Diagram



Step 1

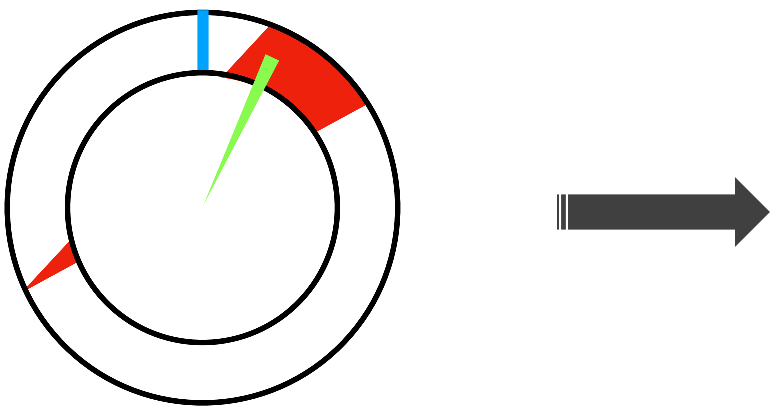
Step 2

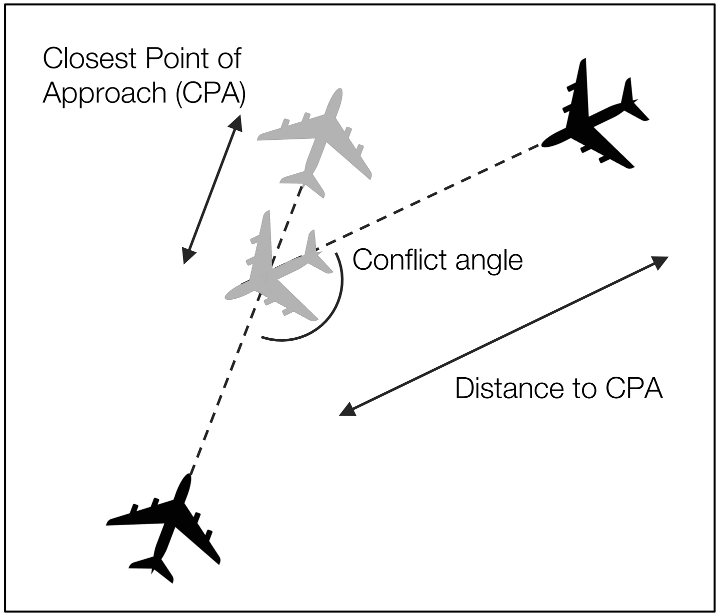
Step 3

TUDelft

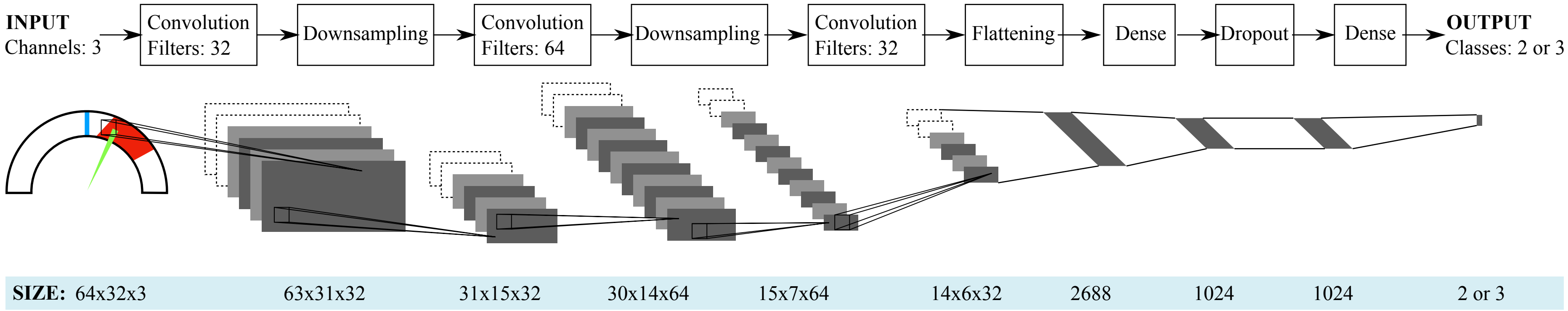
Approach: Choosing the right input

The Solution-Space Diagram

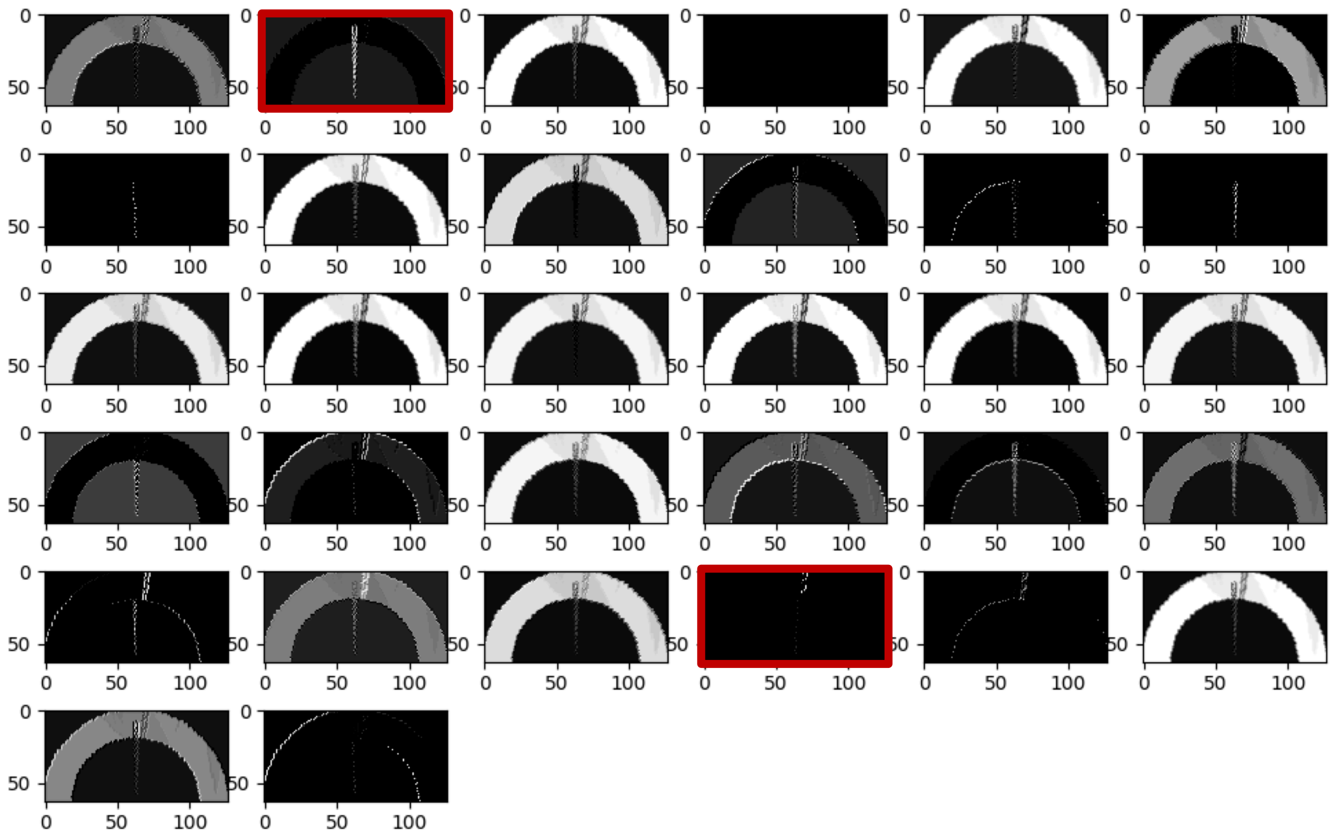




Approach: The model



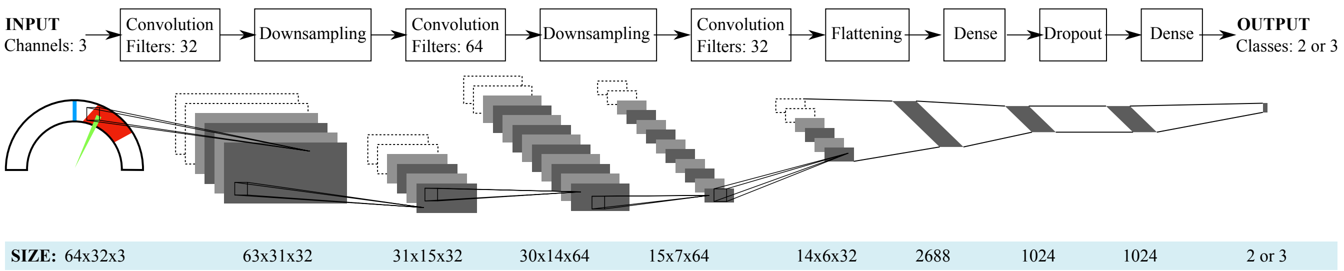
Approach: Example of input layer



Approach: Model output

Three models for three control variables:

1. Resolution type (heading, speed, direct-to)
2. Resolution direction (left/right, up/down)
3. Resolution magnitude



x3

Getting the data: Experiment setup

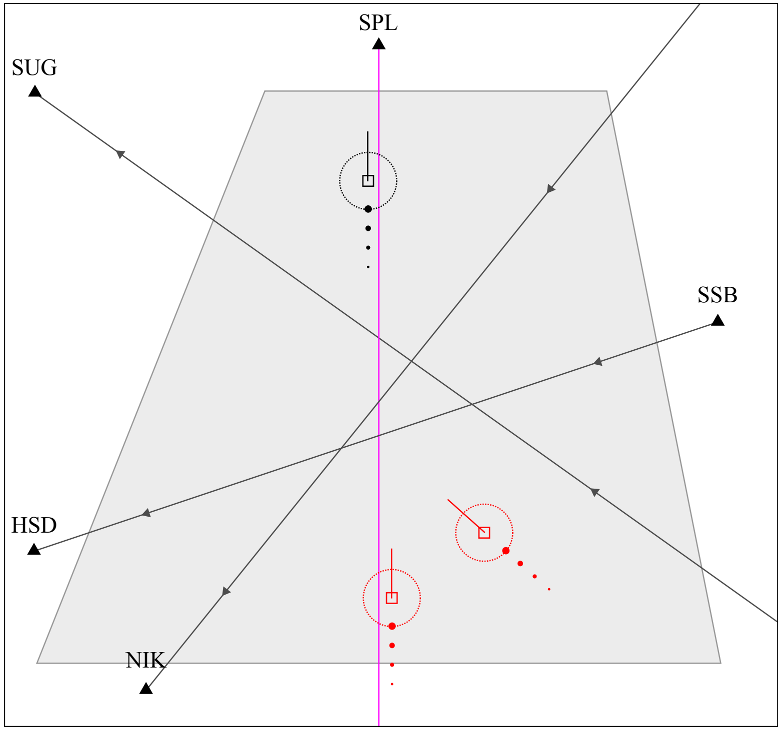
Human in the loop experiment:

Control a sector, while

1. Avoiding Loss-of-Separation between aircraft
2. Guiding the aircraft to their exit waypoint as efficiently as possible

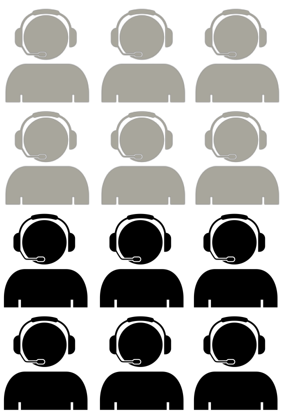
Using either

1. Heading (HDG)
2. Speed (SPD)
3. Direct To (DCT)



Getting the data: Participants and runs

12 participants

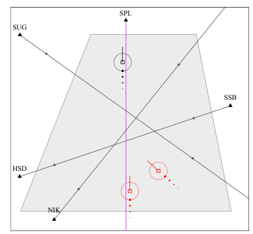


Novice

'Intermediate'

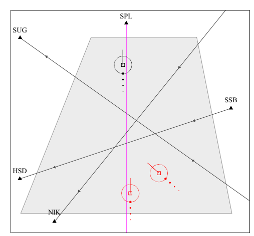
2 scenarios with 4 repeats

scenario 1: 10 conflicts



x 4

scenario 2: 10 conflicts

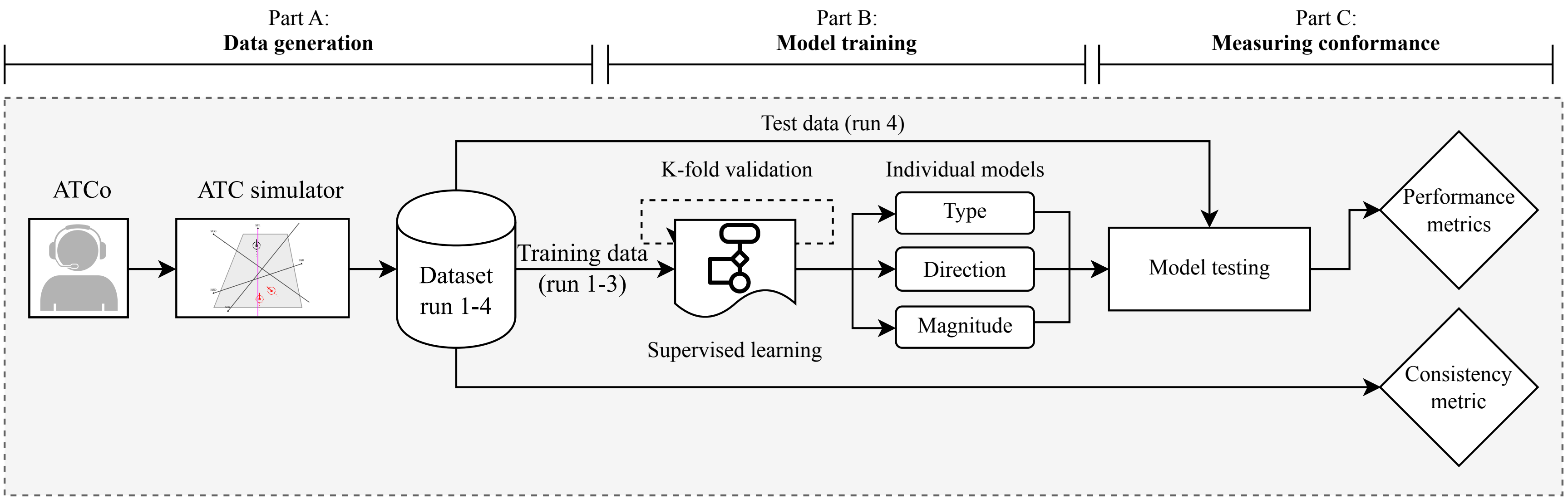


x 4

80 conflicts x 12 participants

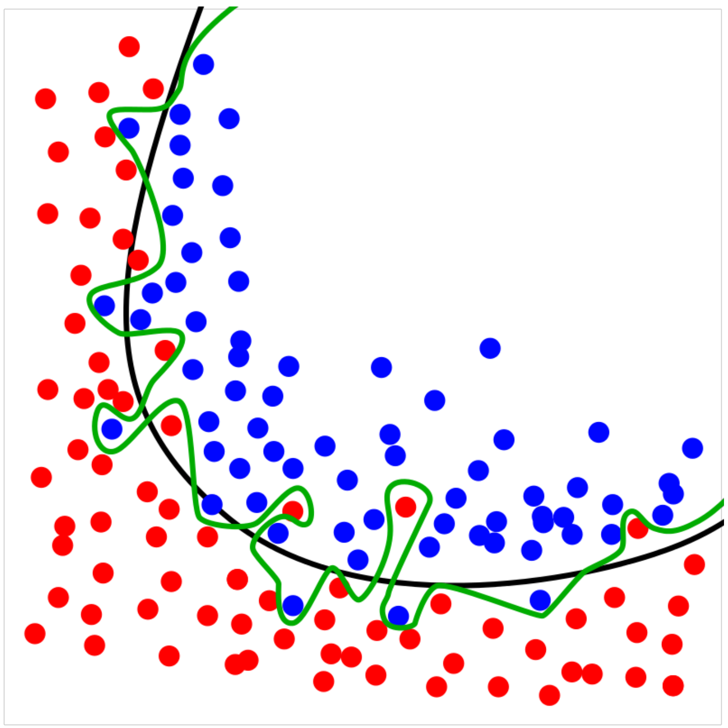
Getting the results: Training models

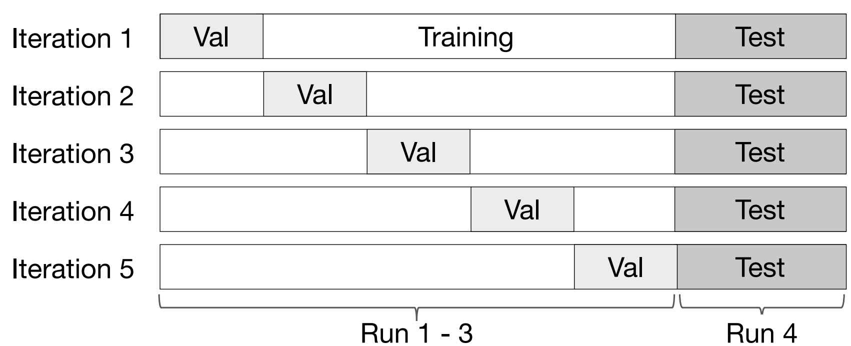
- Three models for type, direction, and magnitude
  - Training data: run 1-3
  - Test data: run 4



Getting the results: Training models

K-fold validation: better performance for small datasets



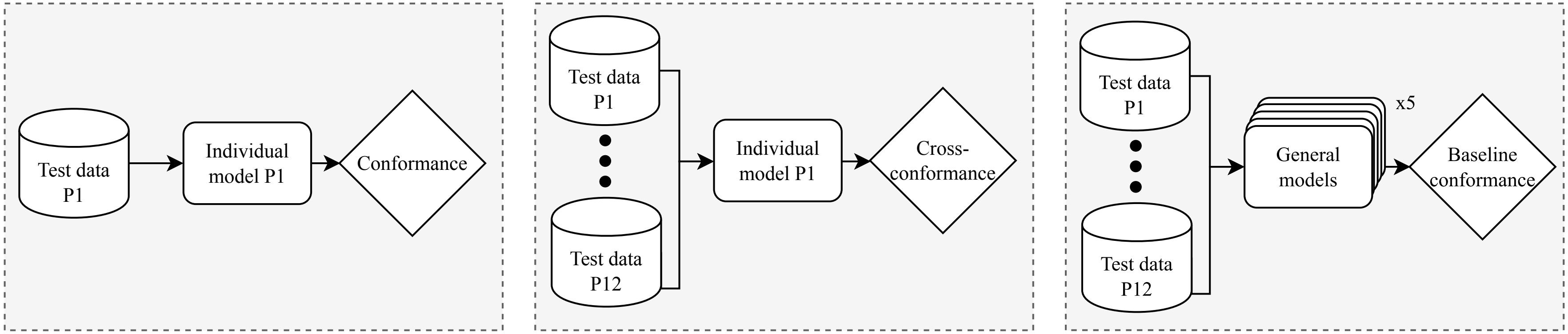


Source: Wikipedia – Overfitting (2019)

Getting the results: Testing conformance

Two types of models: individual and general

Three tests: individual, cross-validation, and baseline



Individual conformance Cross-validation Baseline

Getting the results: Measuring performance

true \ predicted	left	right
left	95	0
right	5	0

Accuracy: 0.95

MCC: 0

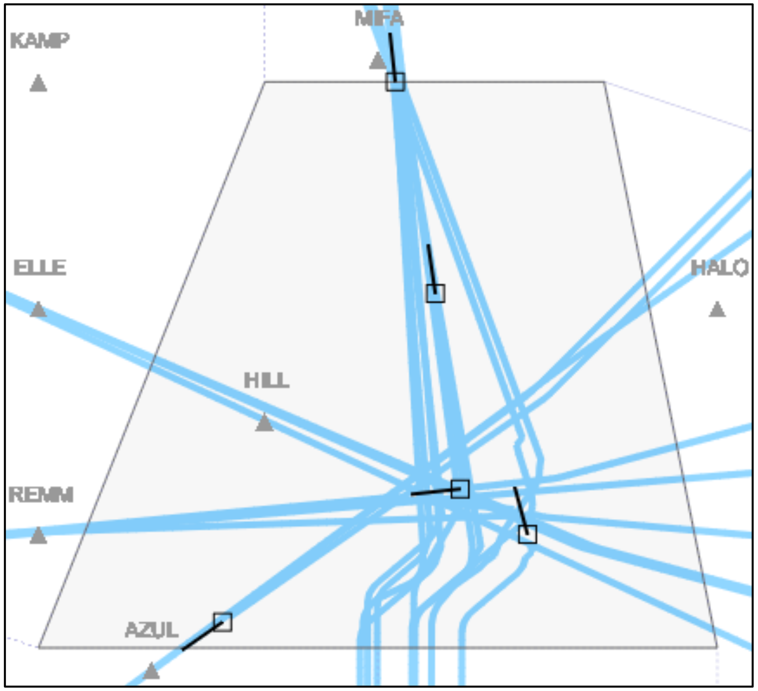
$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

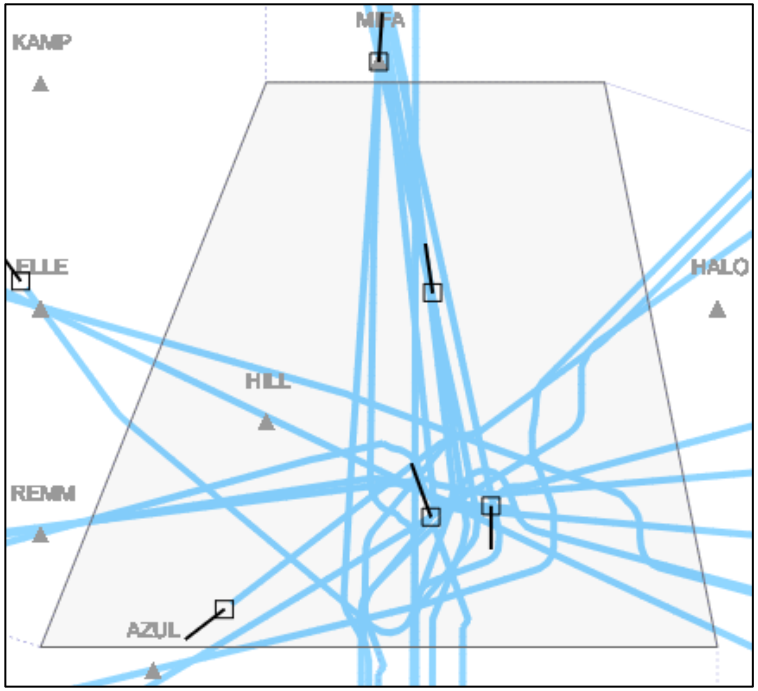
Results

Results: Let's start with some eyeballing

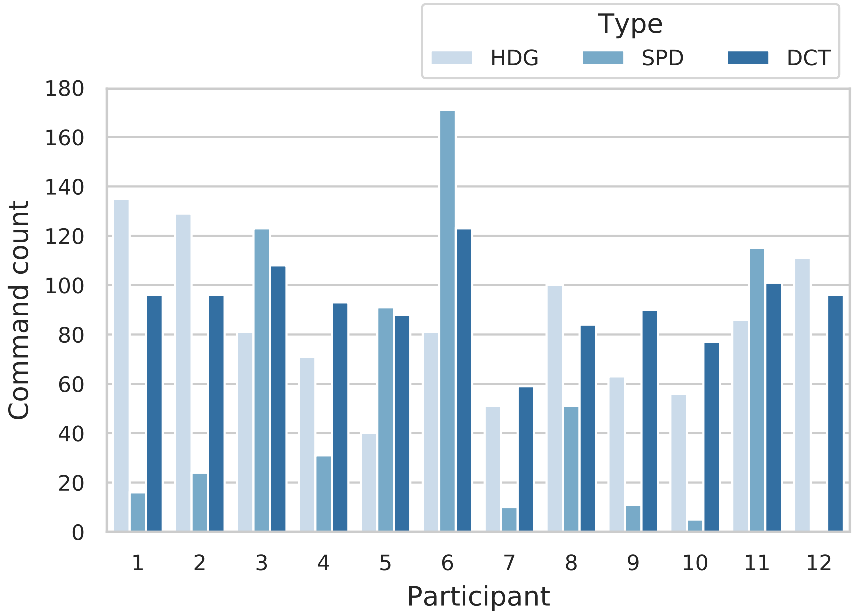
Participant 7, Scenario 1, Run 4

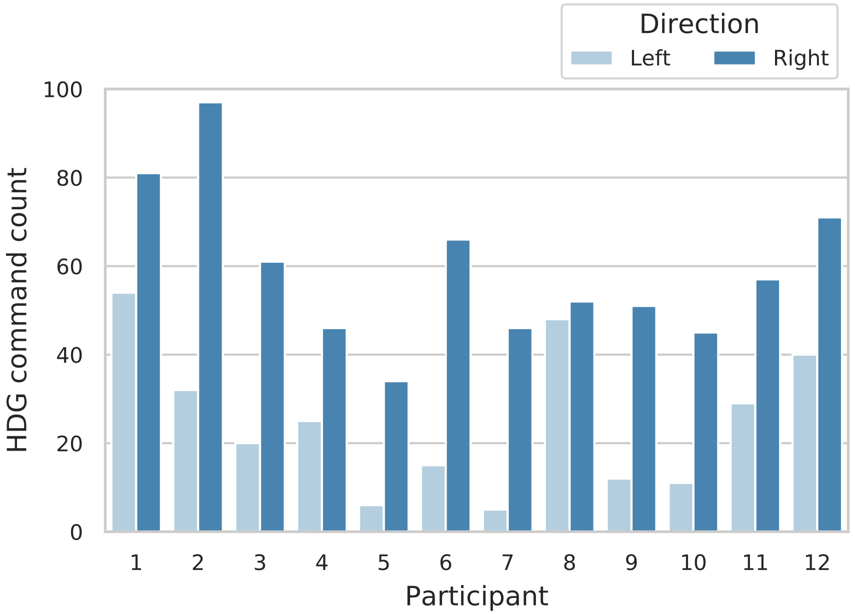


Participant 11 , Scenario 1, Run 4

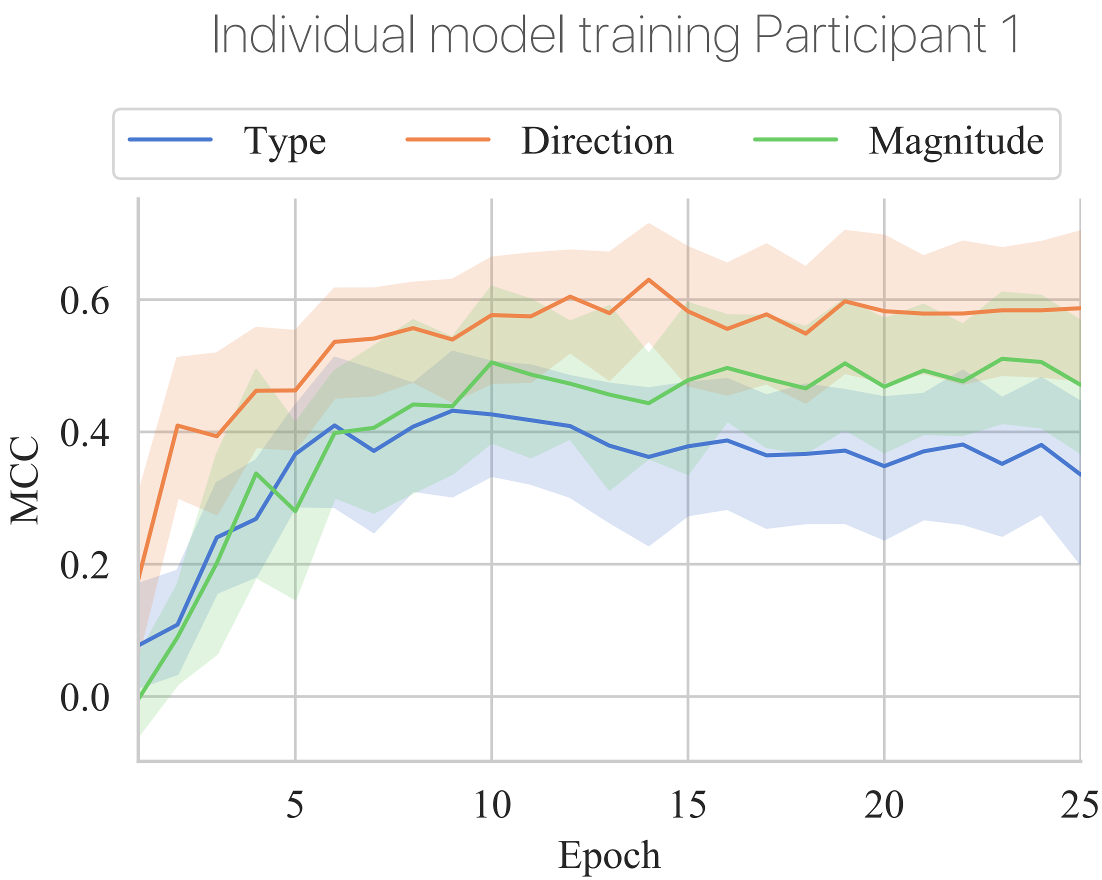


Results: Let’s start with some eyeballing

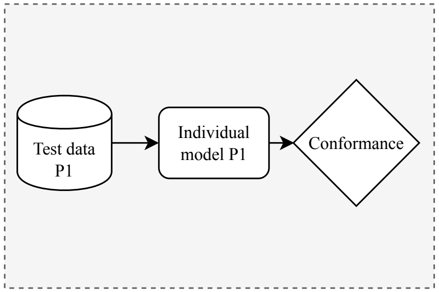


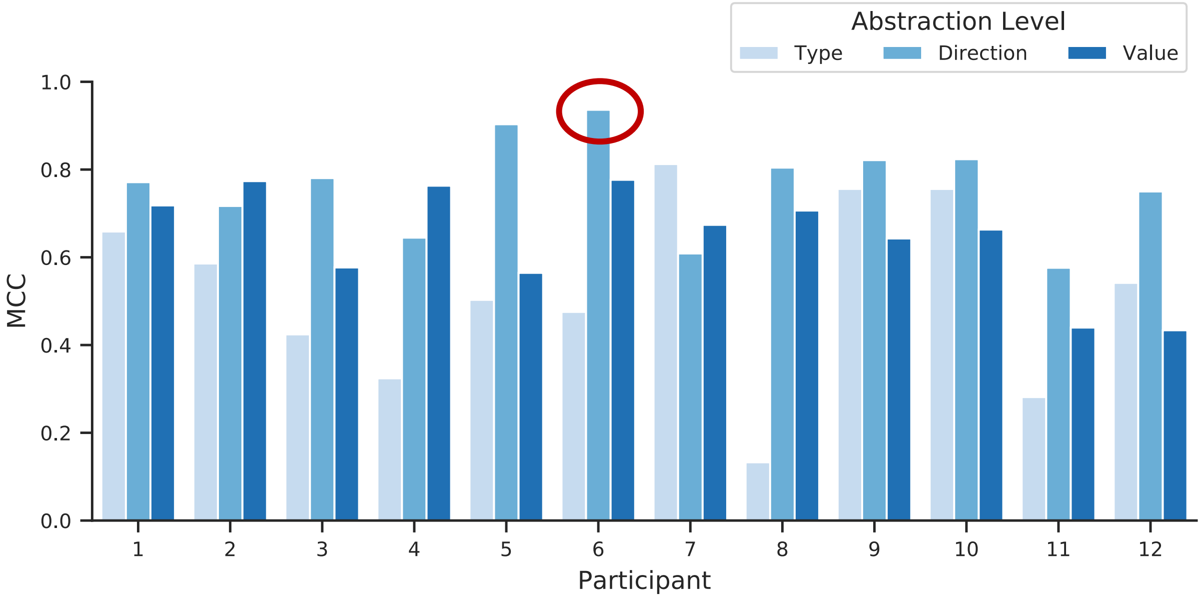


Results: Training performance



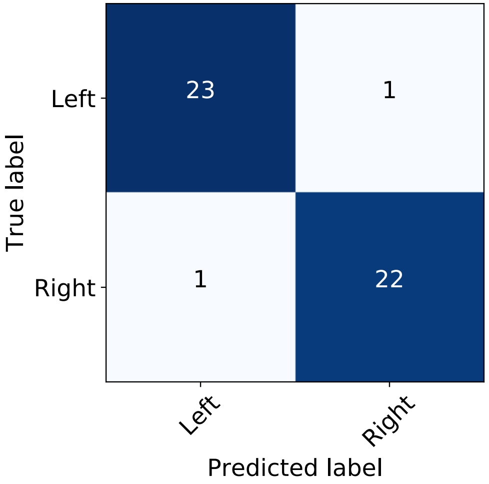
Results: Individual conformance



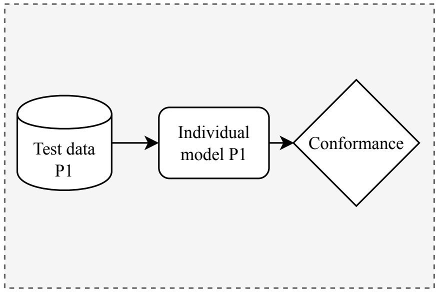


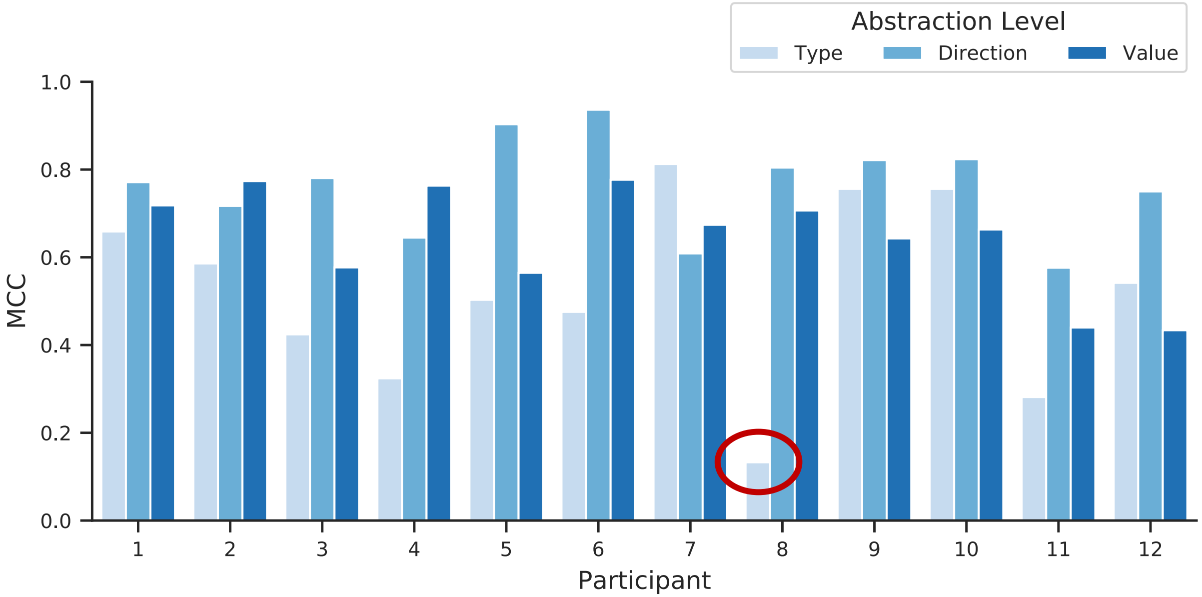
P6 Direction prediction

96% accuracy



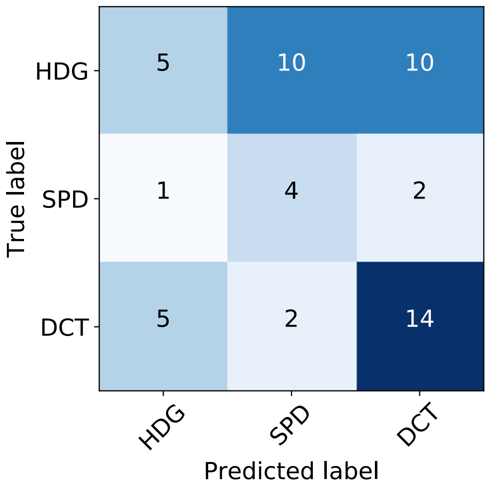
Results: Individual conformance



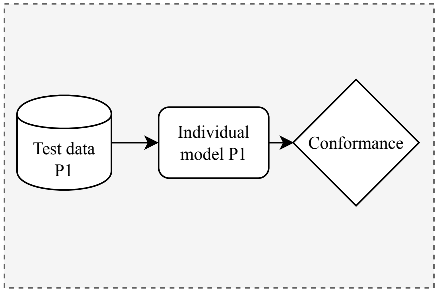


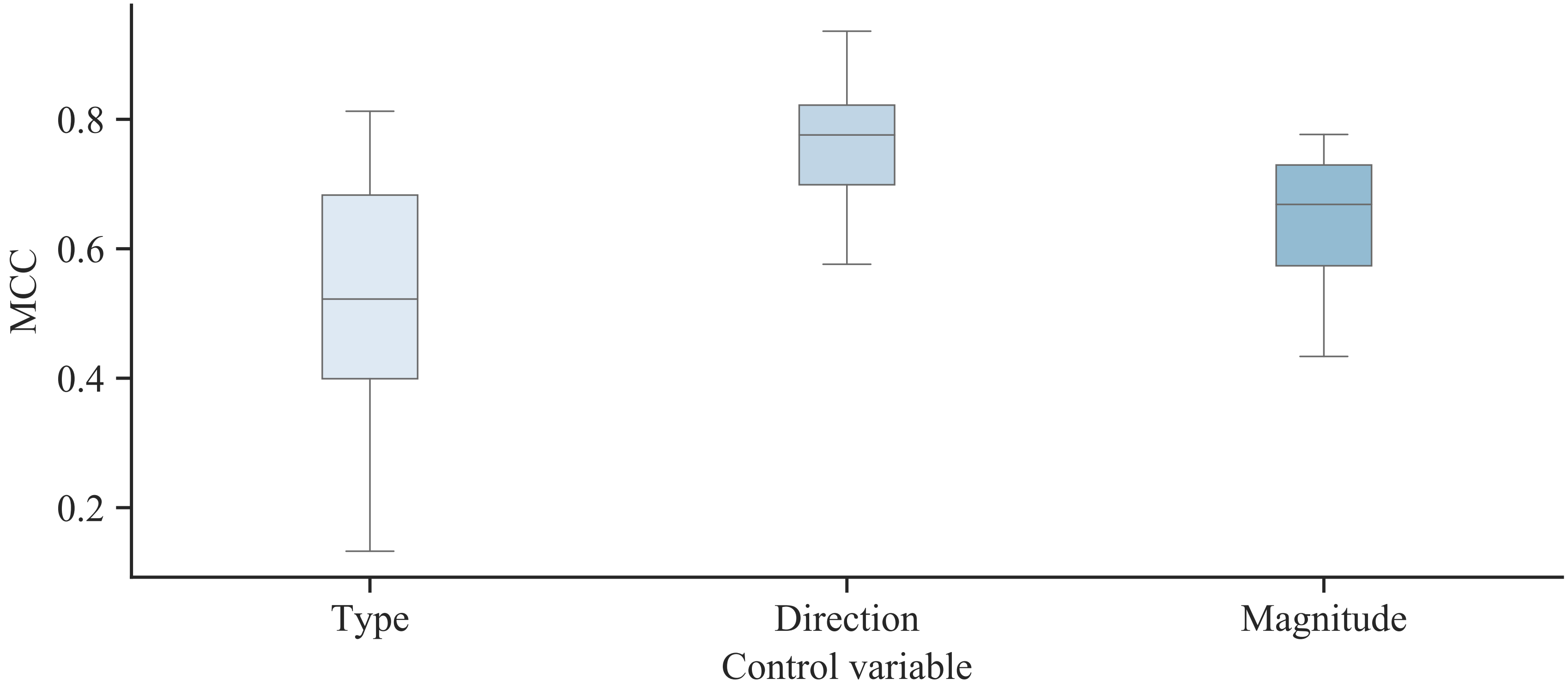
P8 Type prediction

43% accuracy

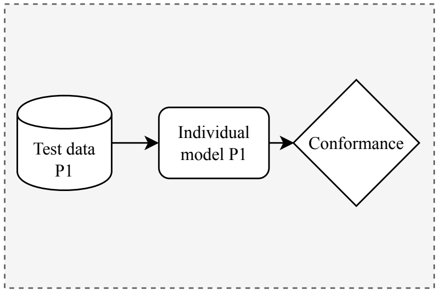


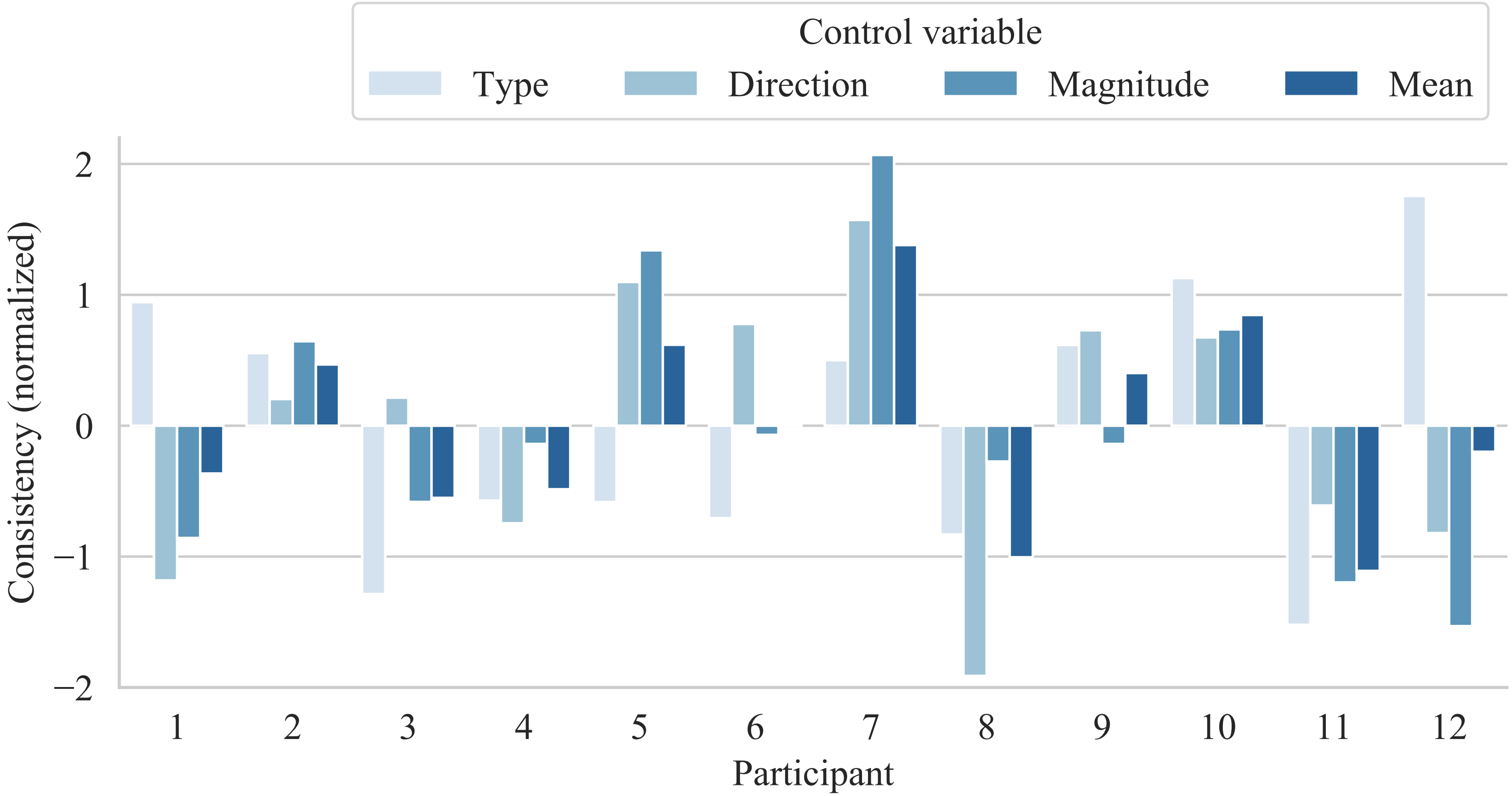
Results: Individual conformance



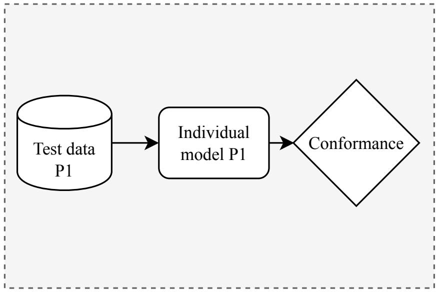


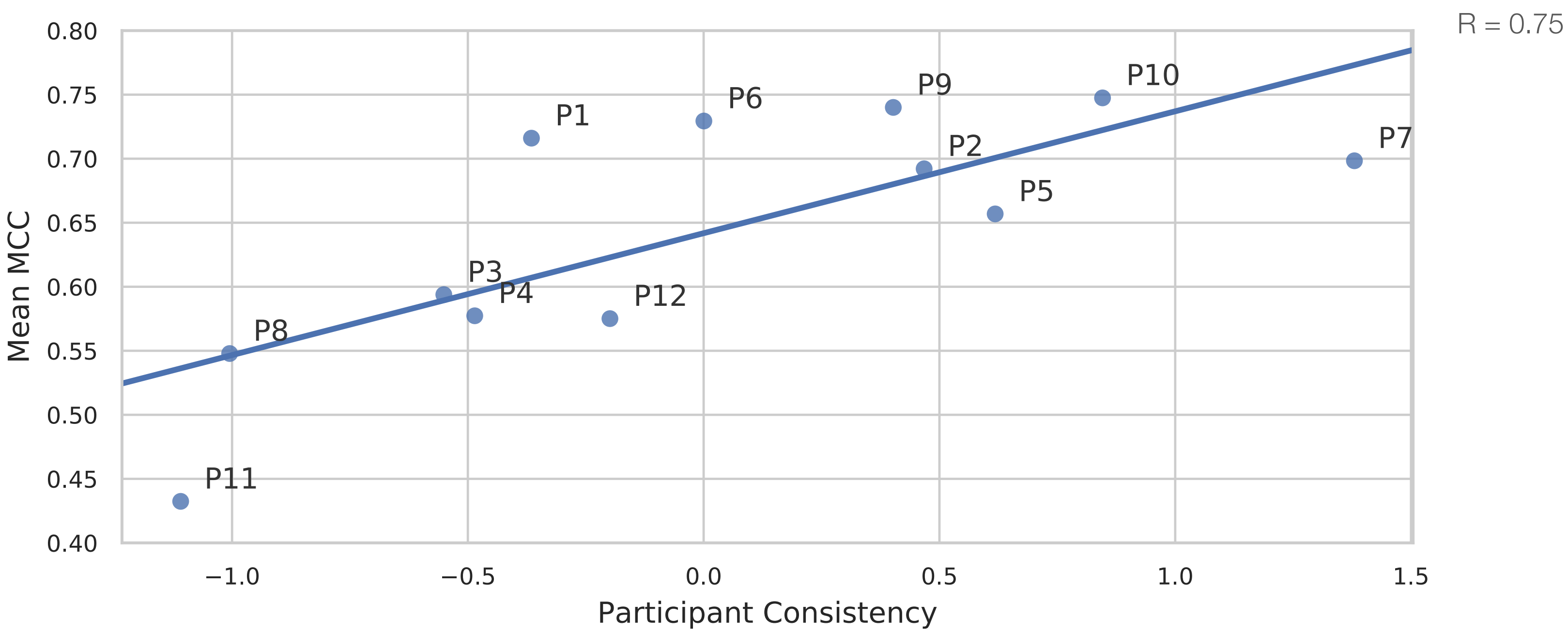
Results: Consistency



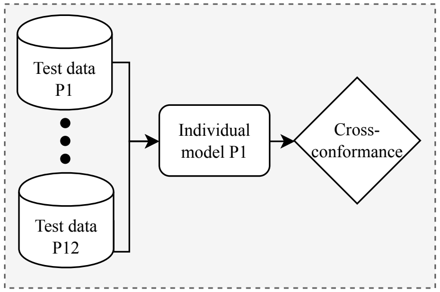


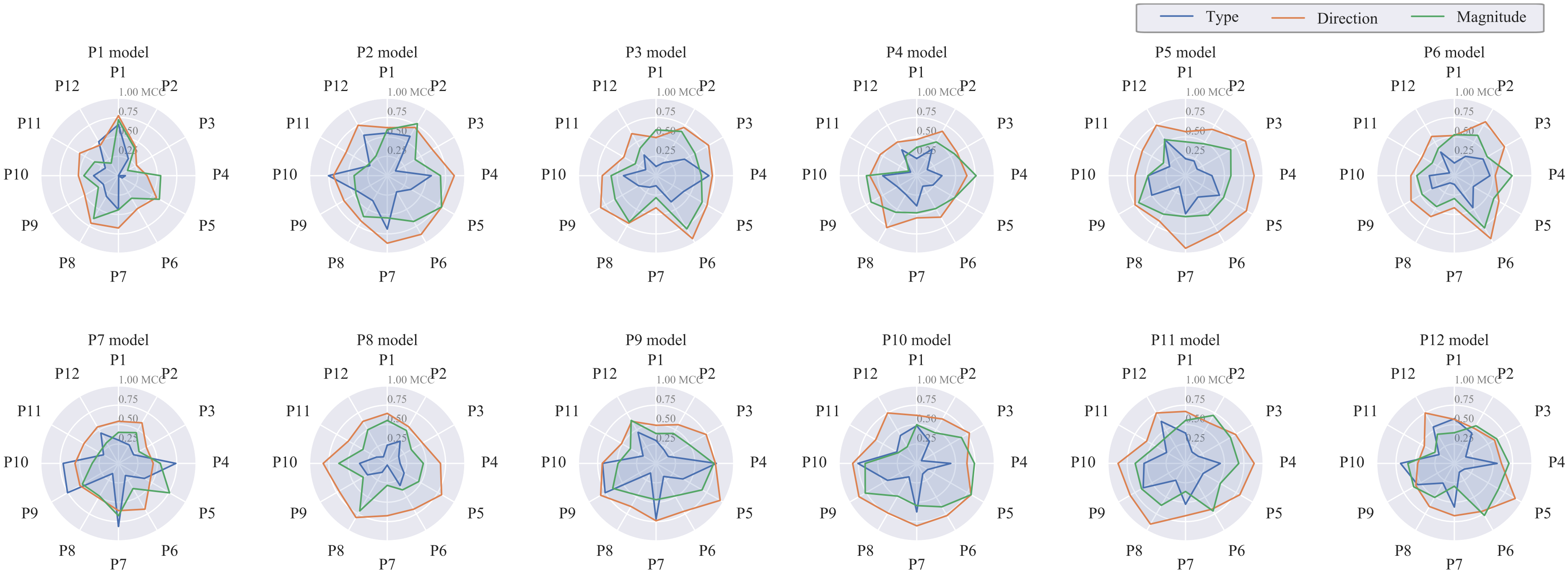
Results: Consistency



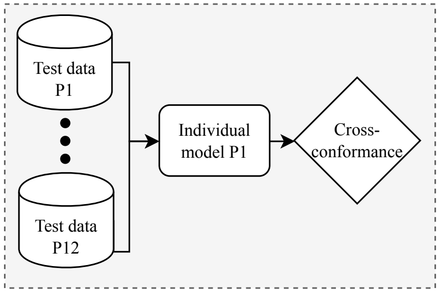


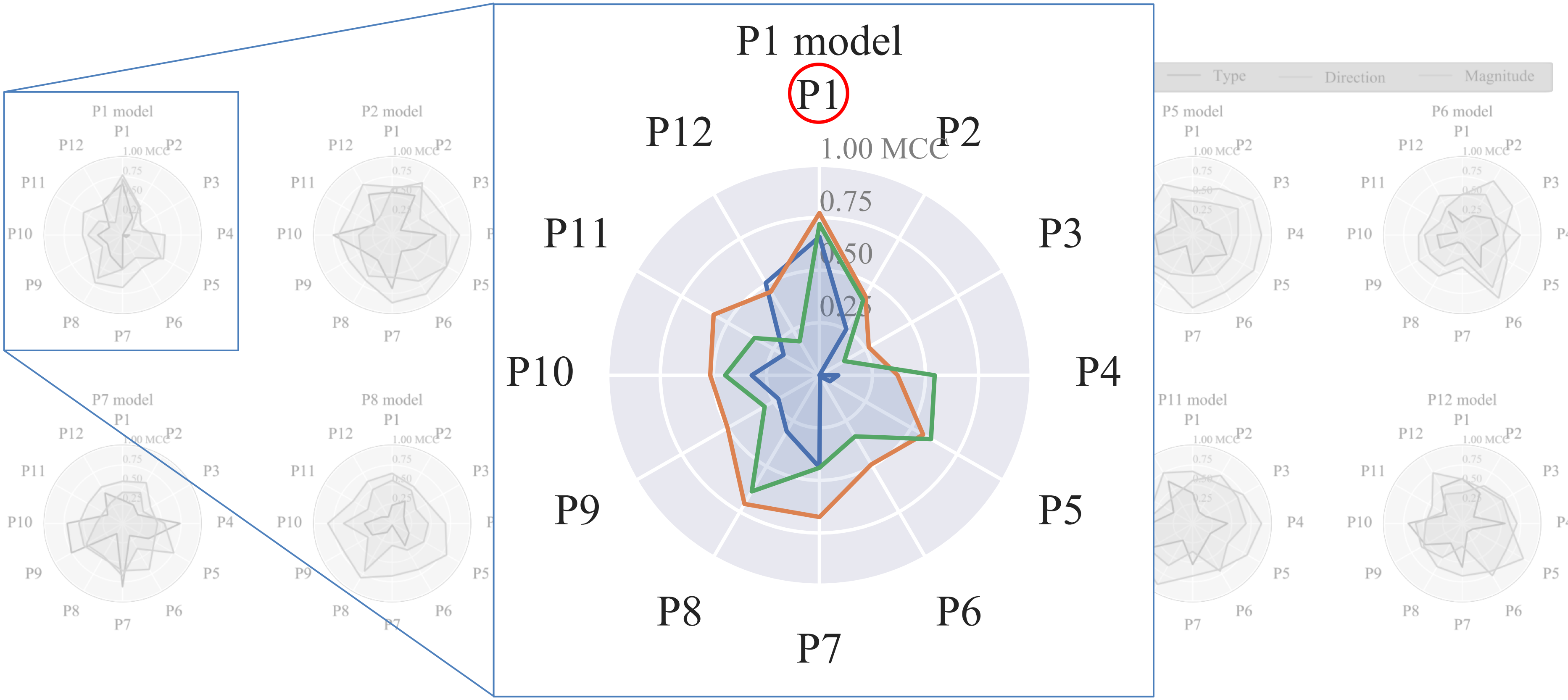
Results: Cross-validation





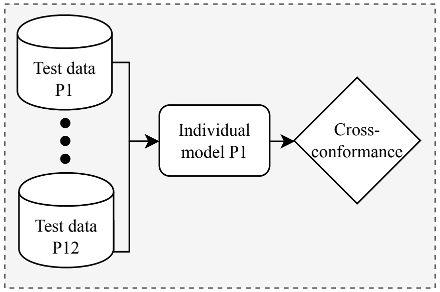
Results: Cross-validation

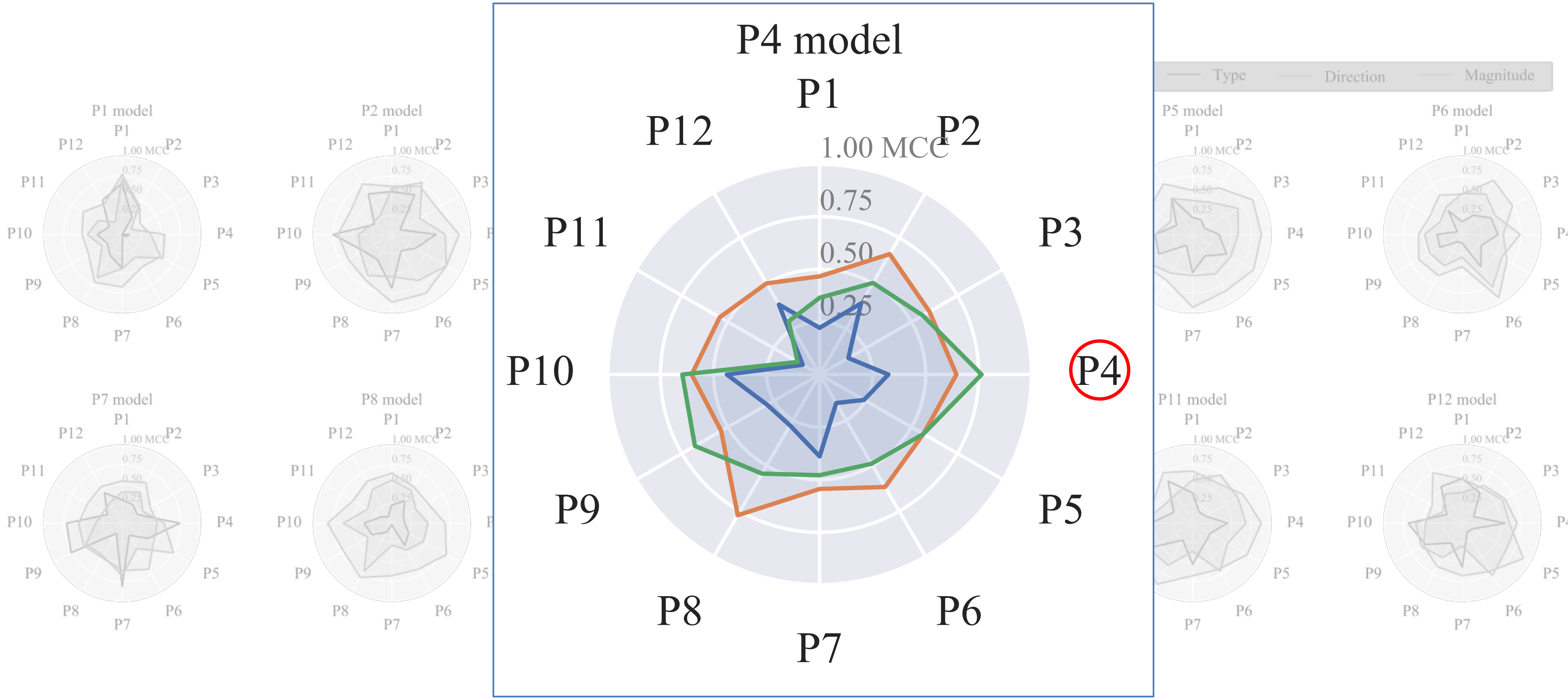




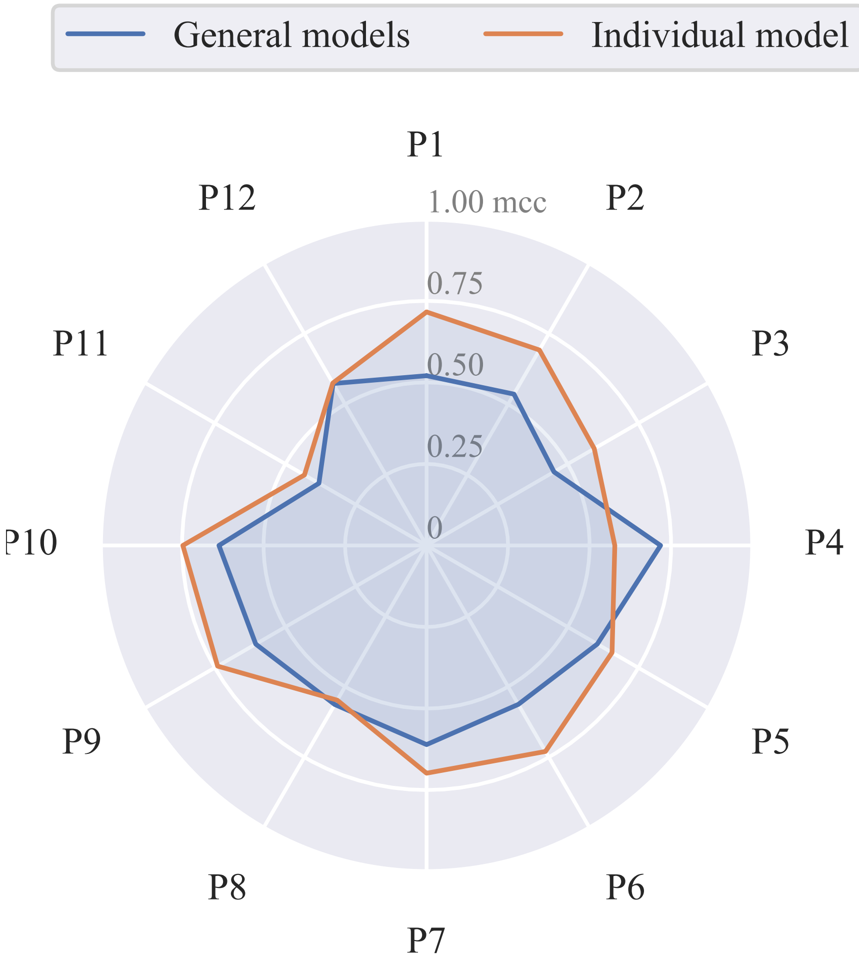
TUDelft

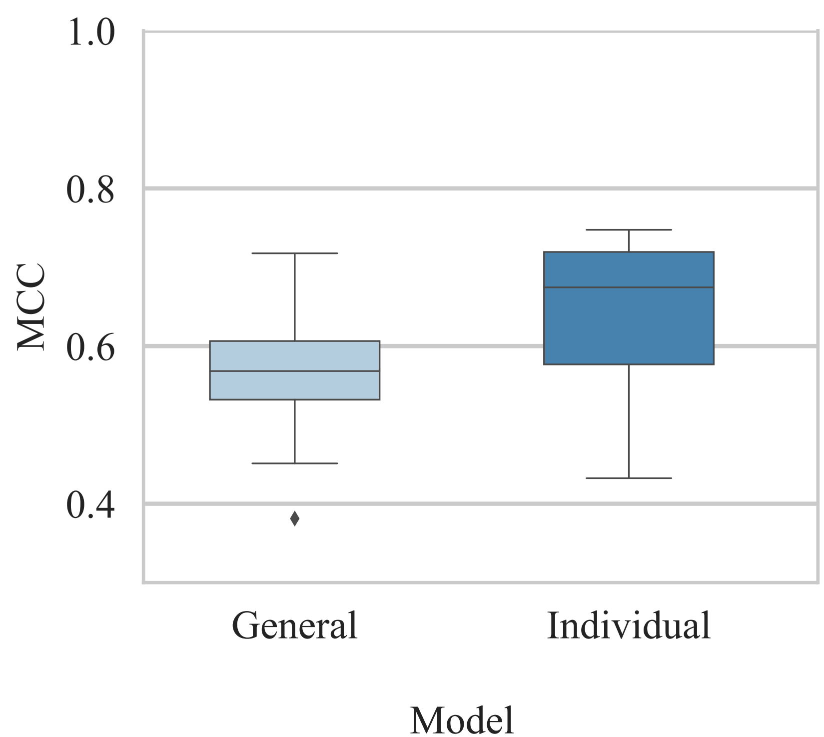
Results: Cross-validation





Results: Baseline validation

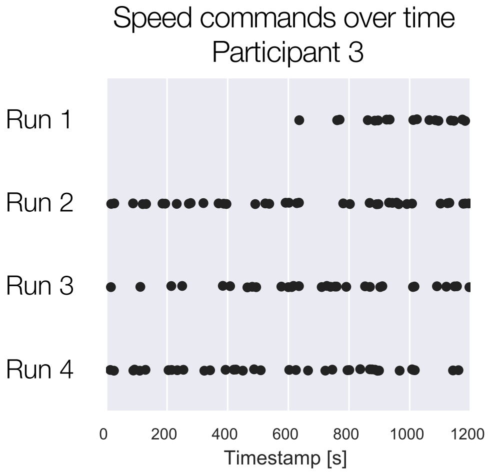




Discussion: Putting things into perspective

Limitations of the experiment

1. Participating 'air traffic controllers' are not professionals
2. Scenarios are constrained in conflict angles and altitude
3. Experiment runs still contain training effects

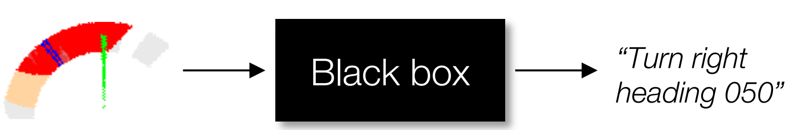


TUDelft

Discussion: Putting things into perspective

Suitability of SSD and machine learning approach

1. Convolutional neural networks converge but overfitting does occur
2. Higher-level decisions and information are not incorporated in the model
3. Neural networks remain a black-box approach



TUDelft

Conclusions: The silver lining

1. SSD images contain sufficient information to predict resolutions in horizontal conflict detection and resolution

2. Convolutional Neural Networks are a feasible approach to achieve individual-sensitive automation

3. Human controllers are sufficiently consistent to train a machine learning algorithm and are strategy heterogeneous as a group

TU Delft

Conformal Automation for Air Traffic Control using Convolutional Neural Networks

J. Ellerbroek

Preliminary analysis

Simulation parameters:

- ATM simulator: BlueSky
- Resolutions: Heading changes only
- Max. number of A/C: Two
- Resolution algorithm: Modified Voltage Potential

BlueSky ATM Simulator

Source: Hoekstra, Ellerbroek - BlueSky ATC Simulator Project: an Open Data and Open Source Approach (2016)

Consistency metric

Type and direction:

$$\text{consistency} = \max\left(\frac{\sum \text{class I}}{\sum \text{class I+II}}; \frac{\sum \text{class II}}{\sum \text{class I+II}}\right)$$

Value:

$$\text{consistency} = \frac{\sum \text{unique values possible}}{\sum \text{unique values used}}$$

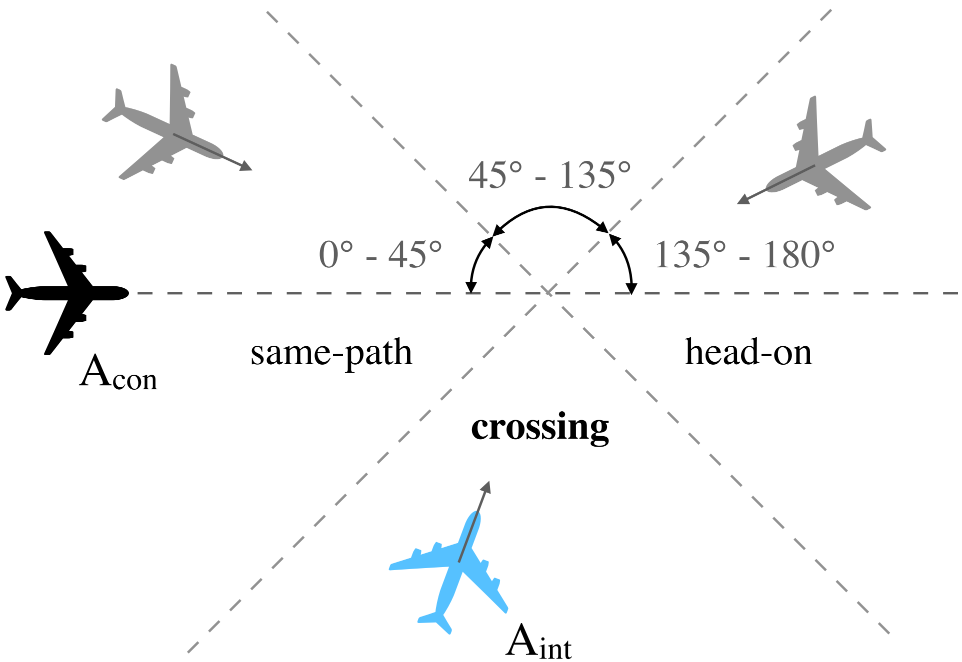
Network Architecture

Layer	Input	Filter size	Stride	Num filters	Activation	Output
Conv2D	32x64x3	2x2	1	32	ReLU	31x63x32
MaxPool	31x63x32					15x31x32
Conv2D	15x31x32	2x2	1	64	ReLU	14x30x64
MaxPool	14x30x64					7x15x64
Conv2D	7x15x64	2x2	1	32	ReLU	6x14x32
Flatten	6x14x32					2688
Dense	2688				ReLU	1024
Dropout	1024					1024
Dense	1024				Softmax	3

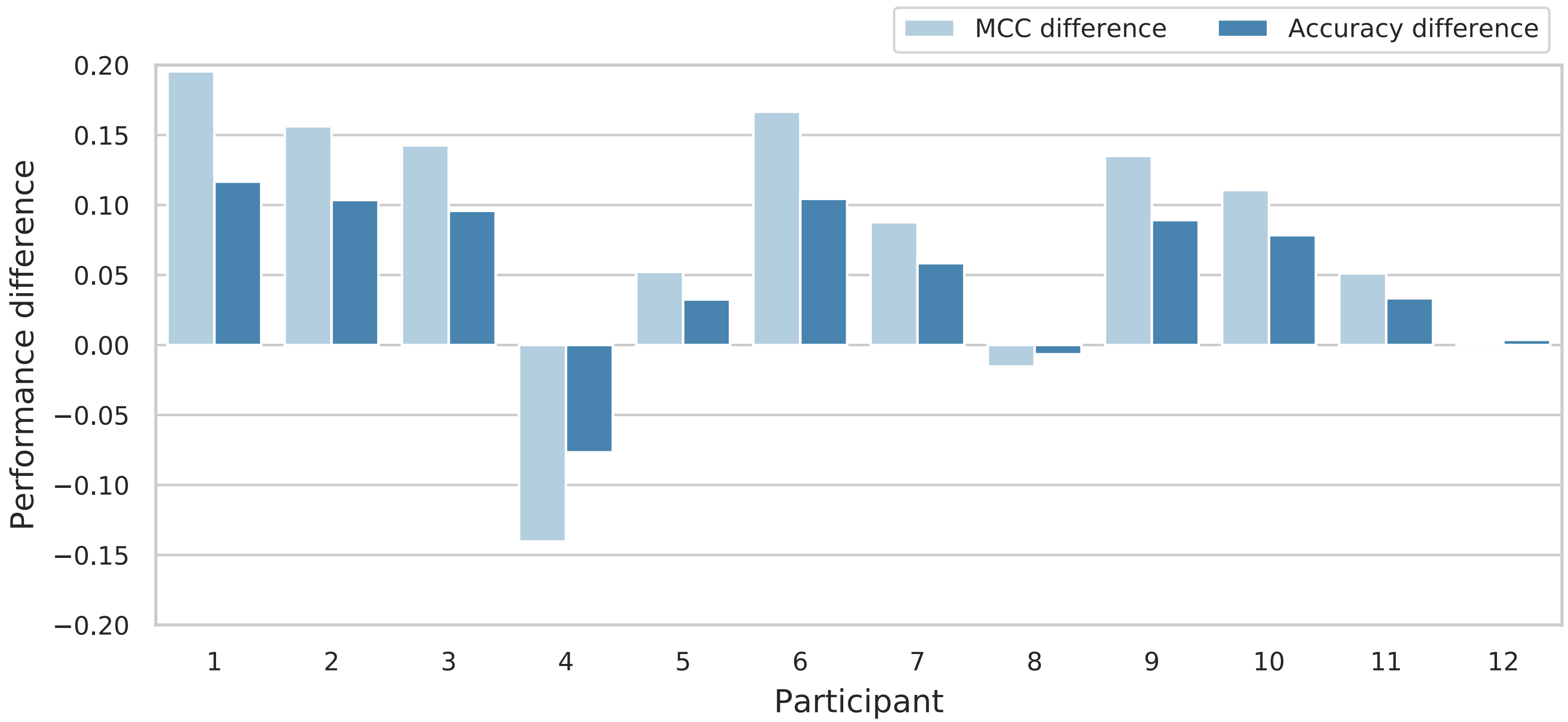
Network Training

Parameters	Value	Unit
Optimization algorithm	Adam	-
Output activation	Softmax classifier	-
Loss function	Categorical entropy	-
Train/val/test ratio	60/15/25	-
K-folds	5	-
Mini batch-size	32	samples
Steps-per-epoch	2 $\times$ training samples / batch-size	samples
Epochs	30	-
Learning rate	0.01	-
Dropout rate	20	%
Input image dimensions	128x128	px

Crossing conflicts



Increase of MCC and accuracy by using individual models



Performance of P4′s model