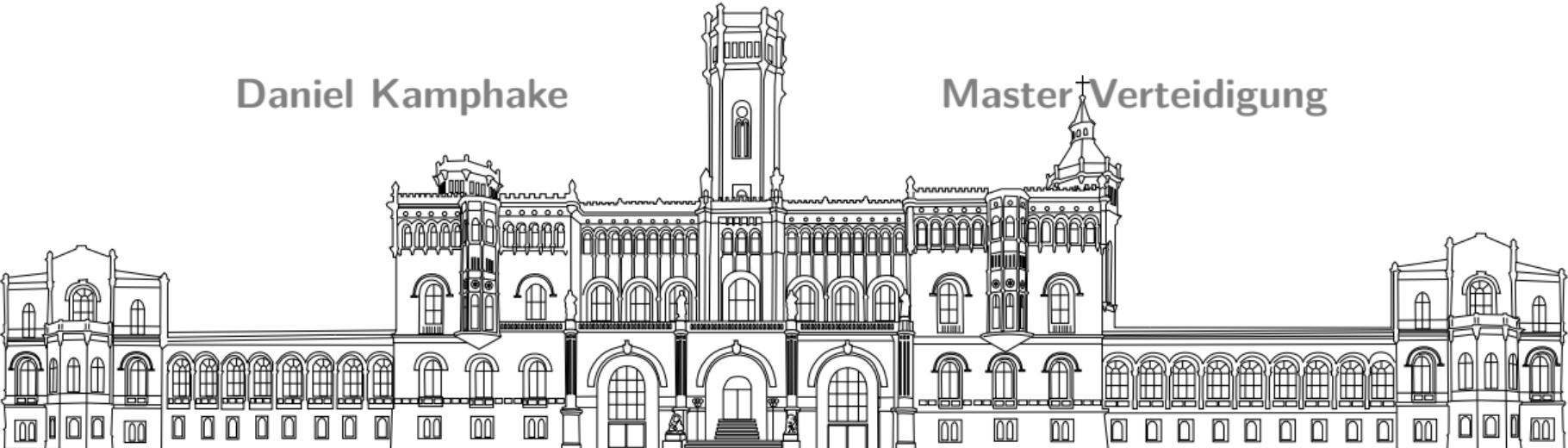


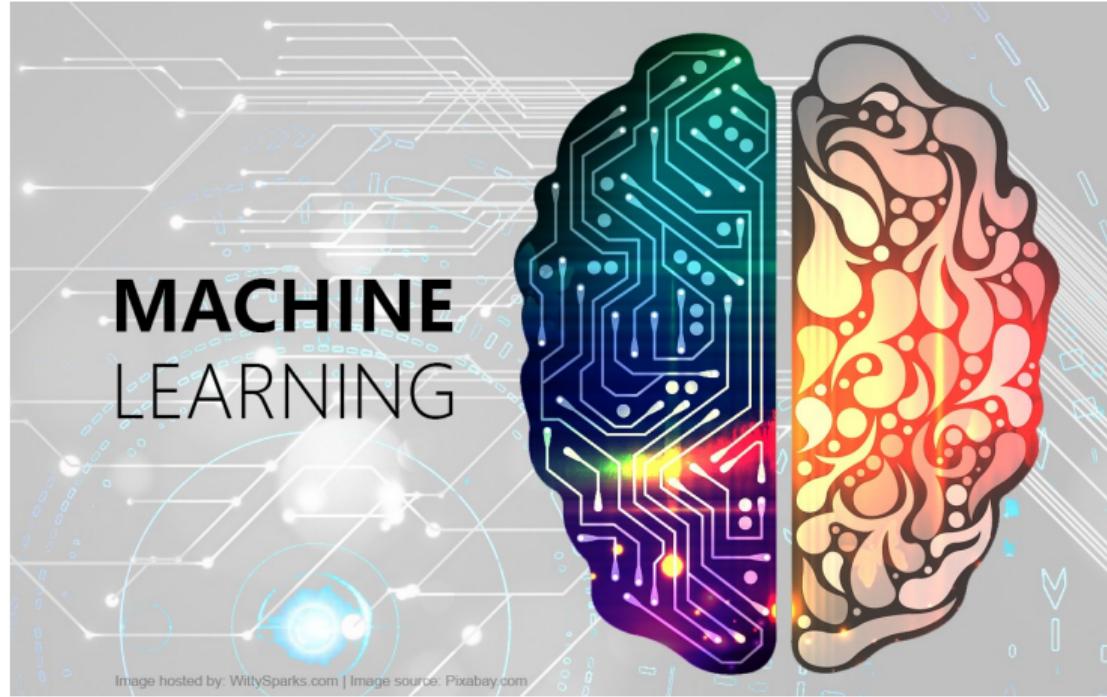
# Optimierung von Neuronalen Netzen für die Analyse des Pupillenreflexes als mobile Diagnoseunterstützung

Daniel Kamphake

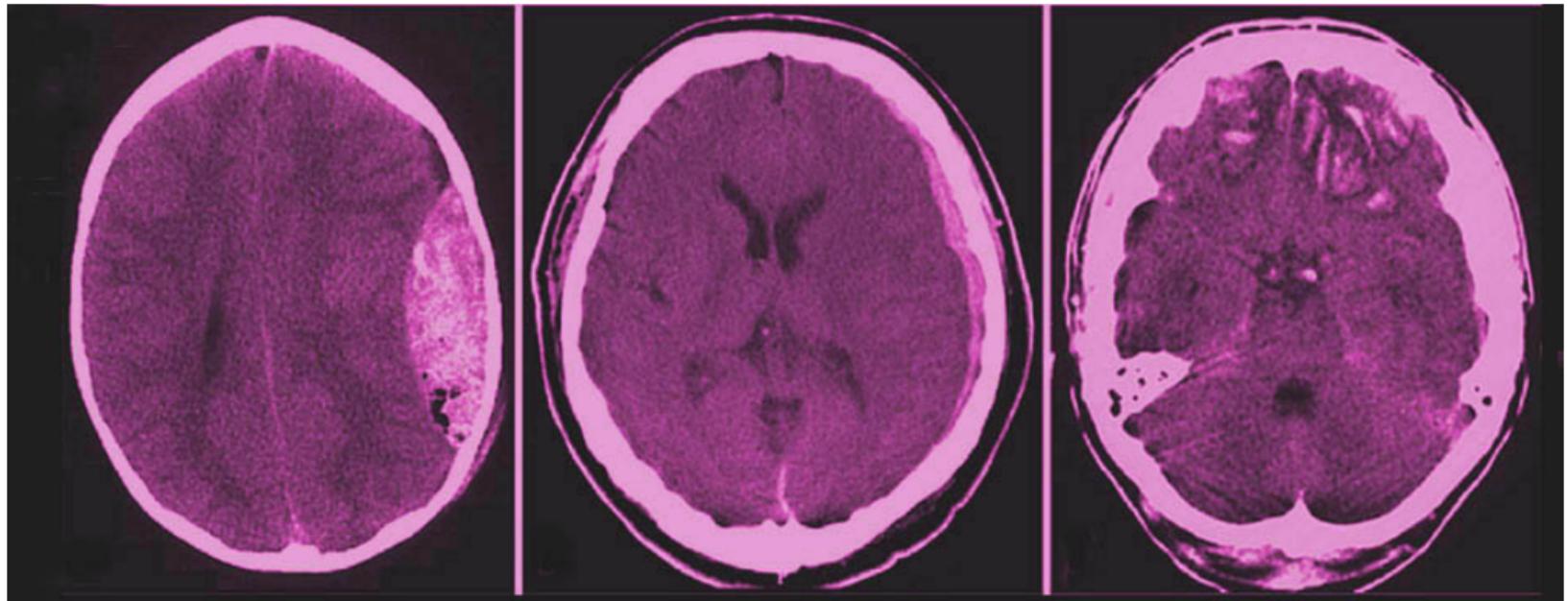
Master Verteidigung



# Motivation



# Motivation



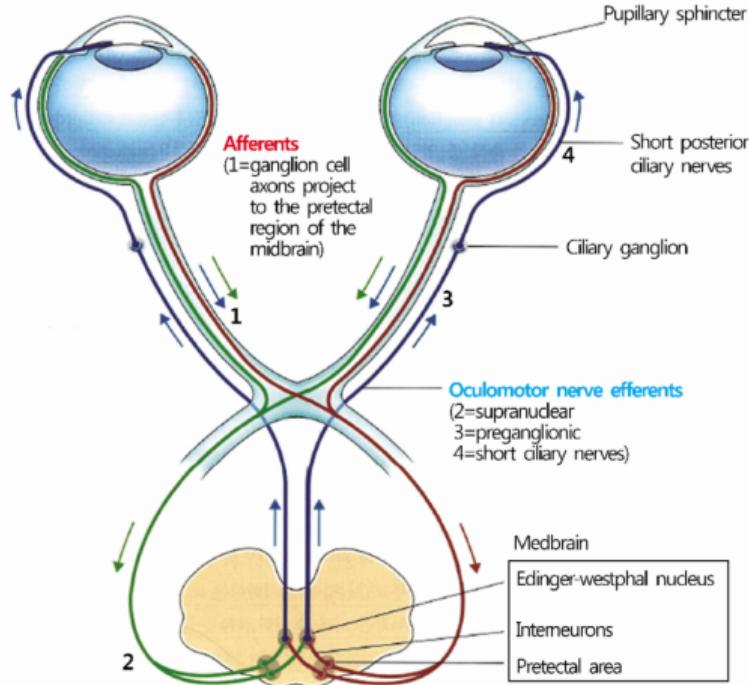
[Els14]



# Grundlagen

Das Auge, Datensätze und neuronale Netze

# Das menschliche Sehvorgang

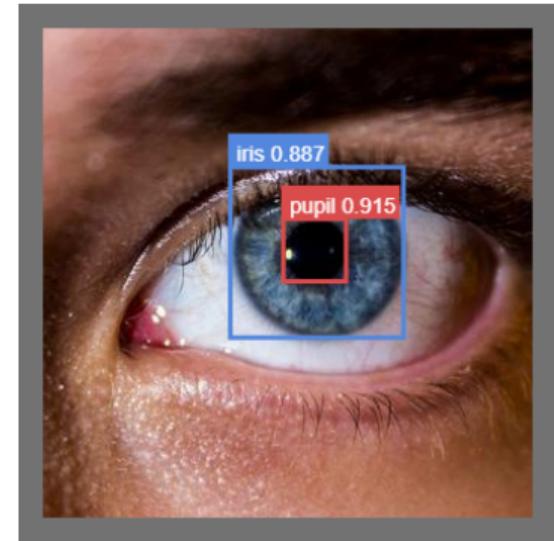
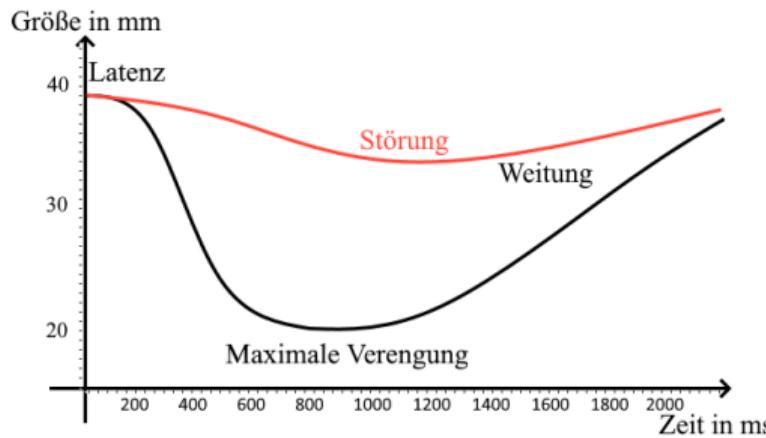


- 1 Aufnahme der Informationen
- 2 Transport über den Nervus Opticus
- 3 Verarbeitung im Hirnstamm
- 4 Signalausgabe im Edinger-Westphal Kern
- 5 Übersetzung durch Ganglion Ciliare
- 6 Kontraktion des Sphincter Pupillae

[Kim17]

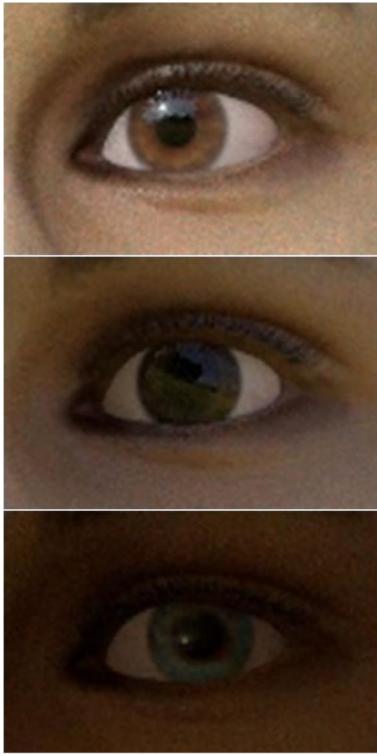
# Der Lichtreflex

- Segmentierung in Pupille und Iris
- Änderungen bei Lichteinstrahlung
- Normaler und beschädigter Verlauf qualitativ eingezeichnet



geändert nach: [Fli]

# Datensätze

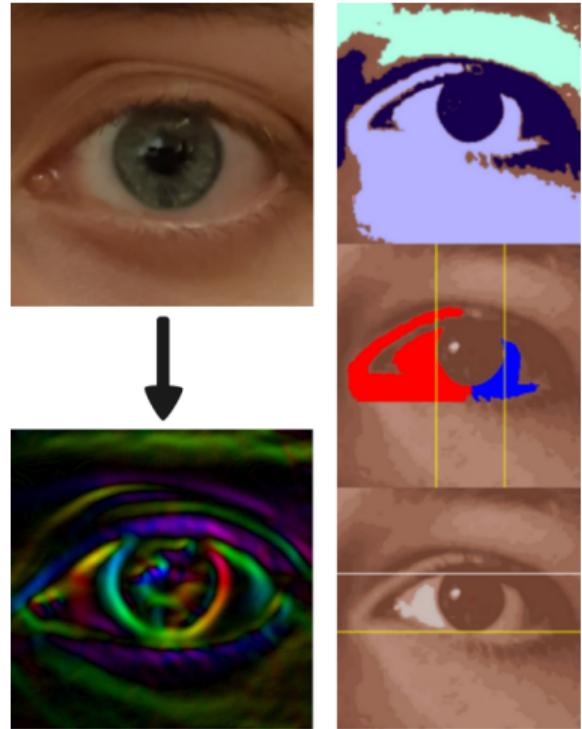


- Vorliegende Datensätze haben eine geringe Auflösung
  - ▣ 80 x 120 Pixel
  - ▣ Prozentual großer Schaden bei kleinen Fehlern
- Wenig Variation im Datensatz
- Keine Reflektionen von Licht
- Umfasst nur ca. 600 Bilder
- Andere Datensätze erfüllen die Ansprüche nicht

Augen: [WBZ<sup>+</sup>15]

# Klassische Methoden

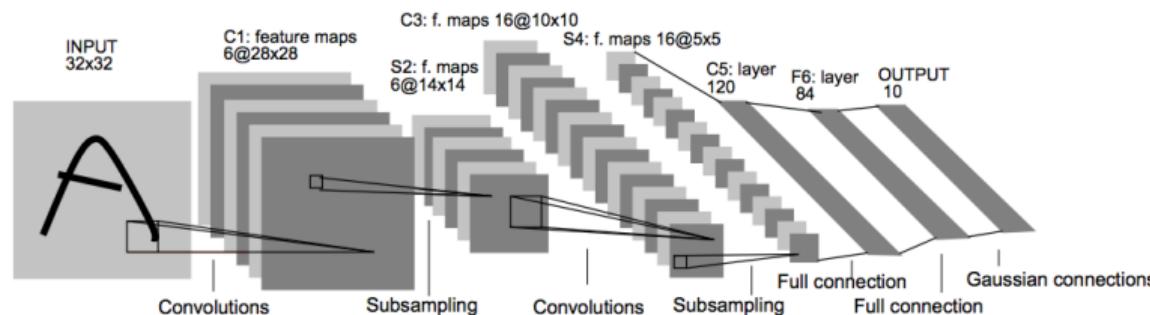
- Die Aufgabenstellung wurde von *Moryäner* [Mor21] bearbeitet
- Klassische Algorithmen sind Zeitaufwendig und ungenau
  - Links: Circular Hough Transformation
  - Rechts: Anpassung von k-Means-Clustering
  - Echtzeitanwendung nicht möglich
- Stattdessen: Neuronale Netze
  - Schneller und Ressourcensparend
  - Benötigen mehr Daten



Geändert nach: [Mor21]

# Neuronale Netze

- Paradigmenwechsel in der Programmierung: Selbstständiges Lernen
- Hohe Genauigkeit bei besserer Laufzeit
- Benötigt viele und vielfältigen Daten
- Anpassung der Modellarchitektur an das Problem



# Objekterkennung — Sliding Window

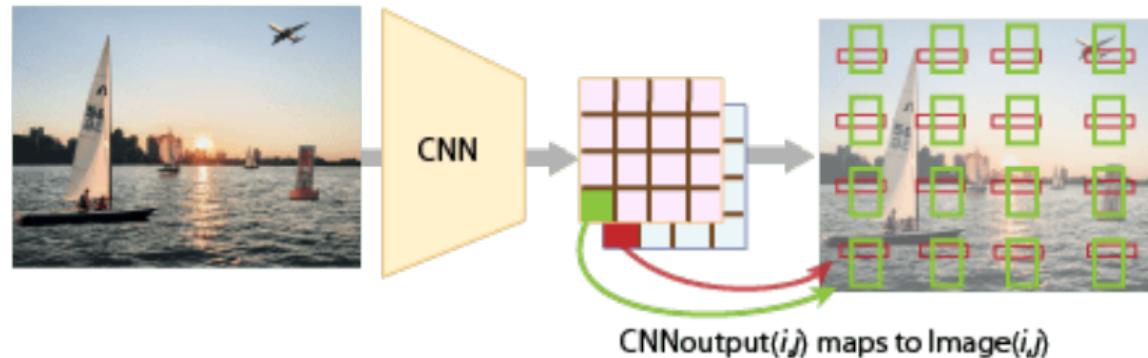
- Für jedes mögliche Fenster wird iterativ eine Klassifikation durchgeführt
- Sehr hohe Komplexität besonders bei vielen Fenstergrößen
- Spätere Verbesserungen mit Region Proposals durch Segmentierung
  - ▣ RCNN, Fast-RCNN, Faster-RCNN



[Rot11]

# YOLO - You Only Look Once

- Alle Erkennungen werden in einer Iteration berechnet
- Einteilung des Bildes in ein Raster
- In jedem Raster befinden sich die gleichen Anchorboxen
- Parallel Erkennung der Anchorbox

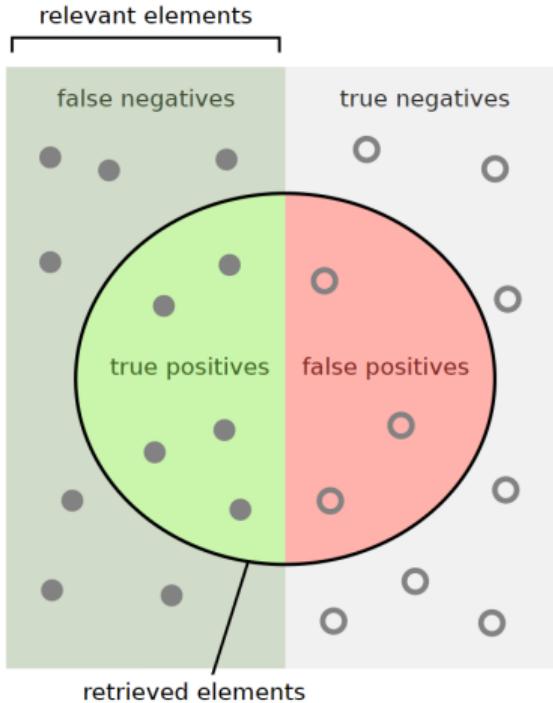




# Optimierung

Bewertung der (Hyper-)Parameter

# Precision und Recall

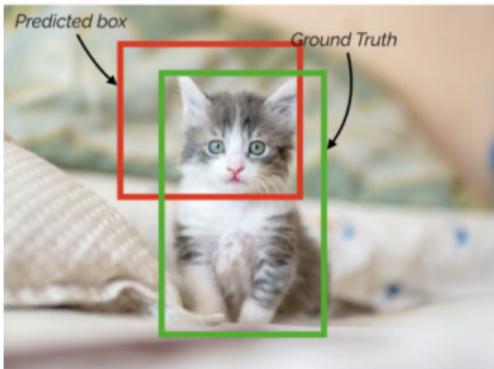


- Precision: Wie viele gefundenen Pupillen sind korrekt
- Recall: Wie viele der Pupillen wurden gefunden
- $F_\beta$ -Score: Optimum der Metriken mit Gewichtung  $\beta$ 
  - Hier: 
$$F_1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

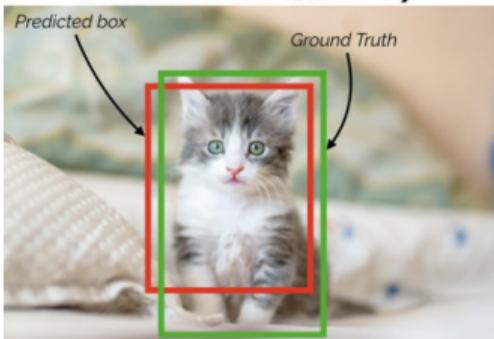
Geändert nach: [Wal14]

# Mean Average Precision (mAP)

False Positive (FP) IoU = ~0.3



True Positive (TP) IoU = ~0.7



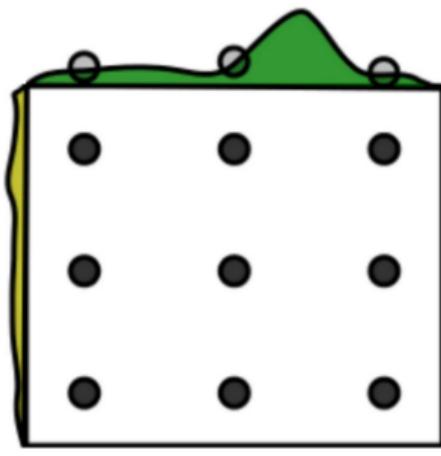
- Die Intersection over Union (IoU) beschreibt die Überlagerung von Ist- und Soll- Bounding Box
- mAP beschreibt die Fläche unter der Precision-Recall Kurve für verschiedene IoU Schwellwerte
- $mAP.95 = \frac{1}{10} \sum_{n=0}^9 Precision_{IoU=50\%+n\times5\%}(Classifier)$

Geändert nach: [YOH]

# Suchalgorithmen

Grid Layout

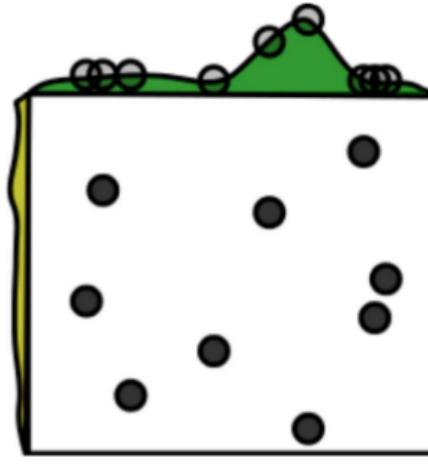
Unimportant parameter



Important parameter

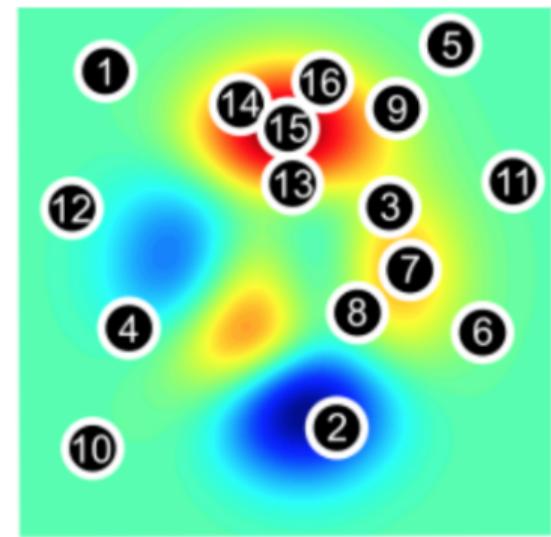
Random Layout

Unimportant parameter



Important parameter

Bayes'sche Suche



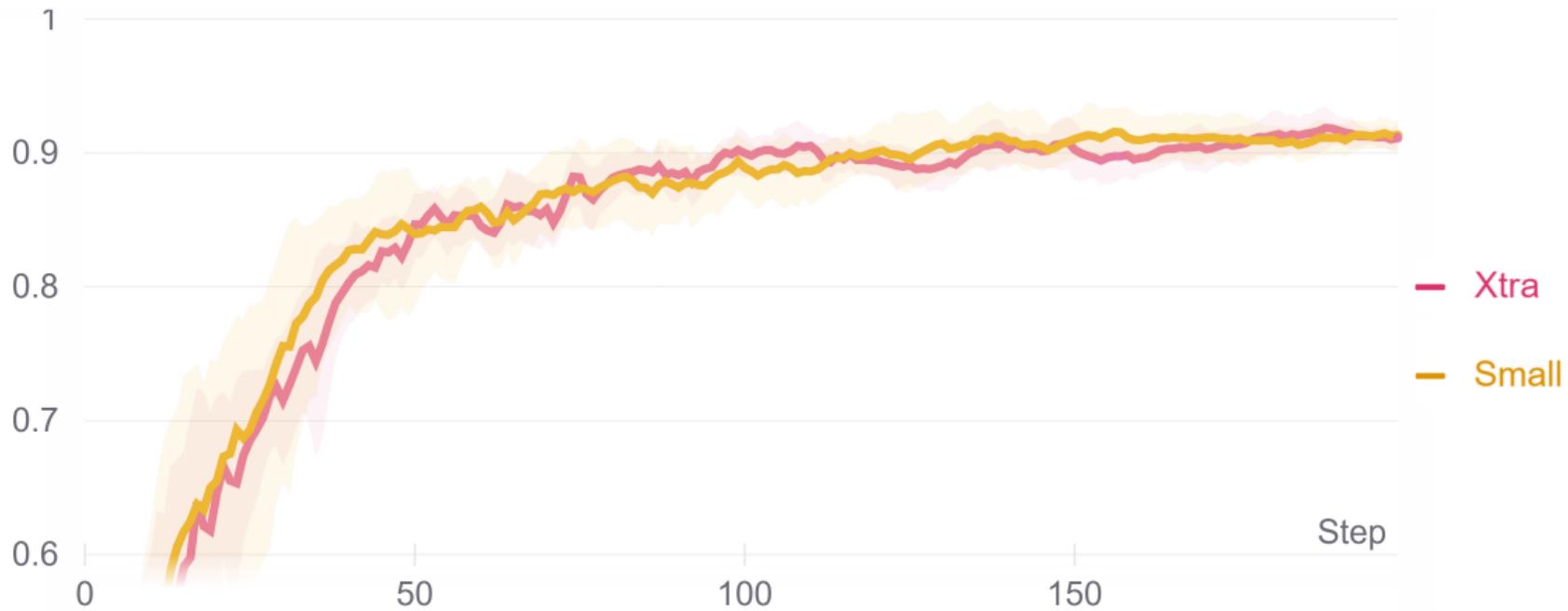
Geändert nach: [BB12] und [LJD<sup>+</sup>17]



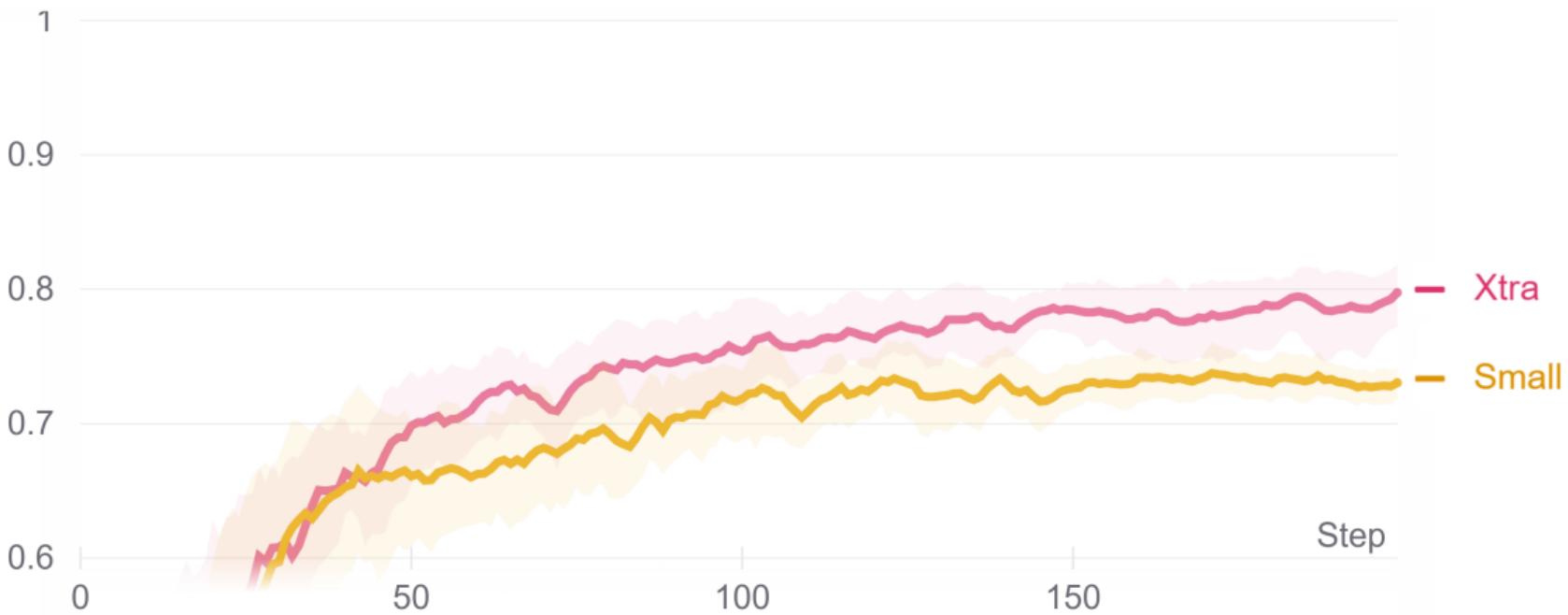
# Unity Eyes

Erstellung eines Datensatzes

# Bestehende Datensätze — Iris mAP.95

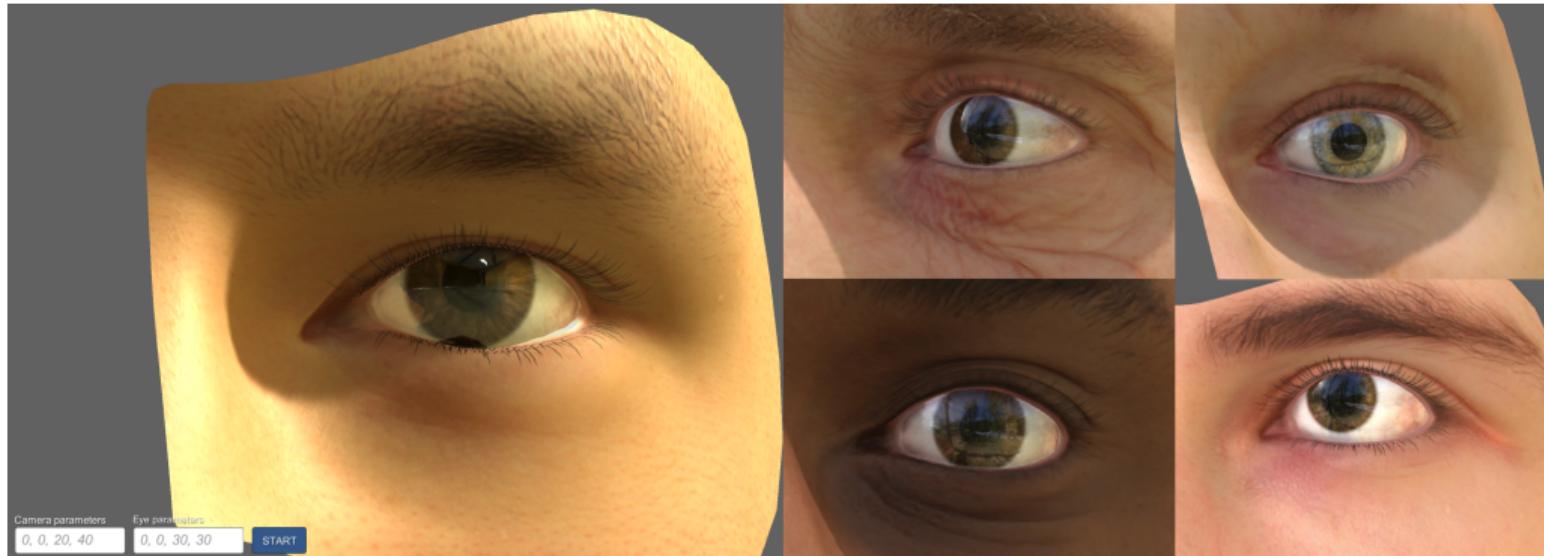


# Bestehende Datensätze — Pupille mAP.95

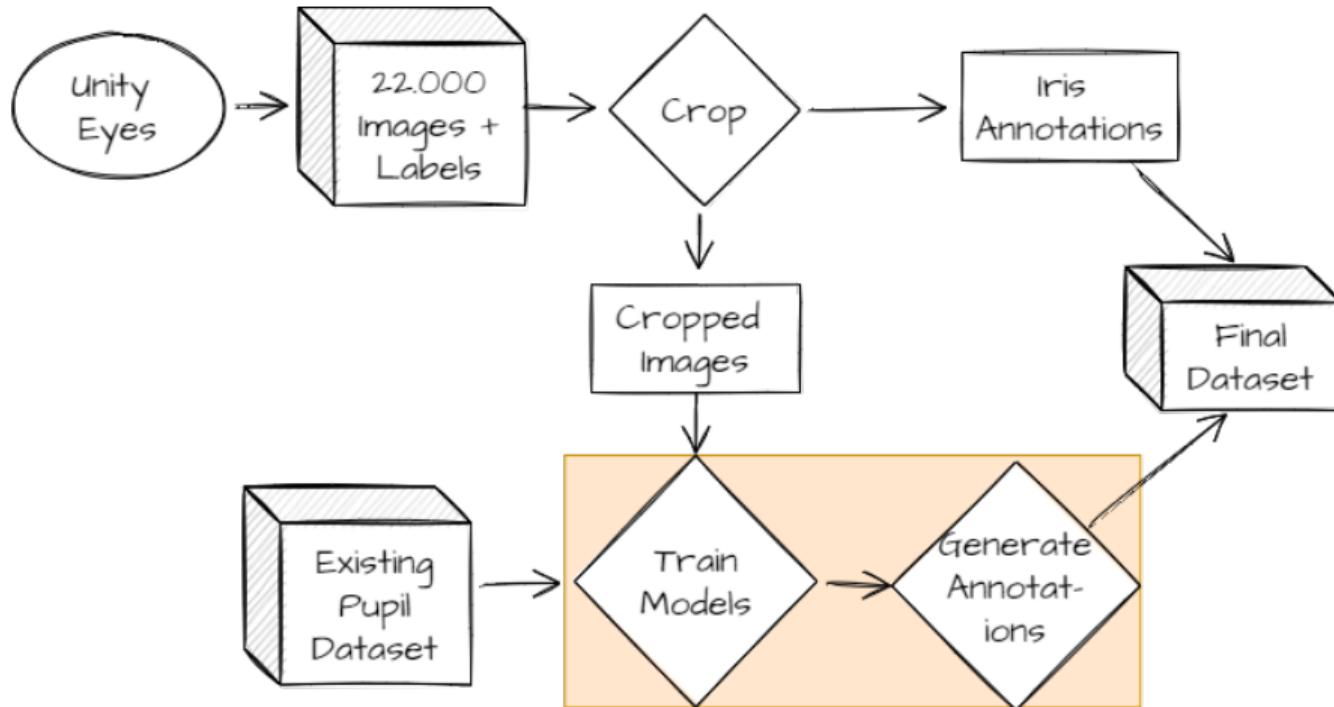




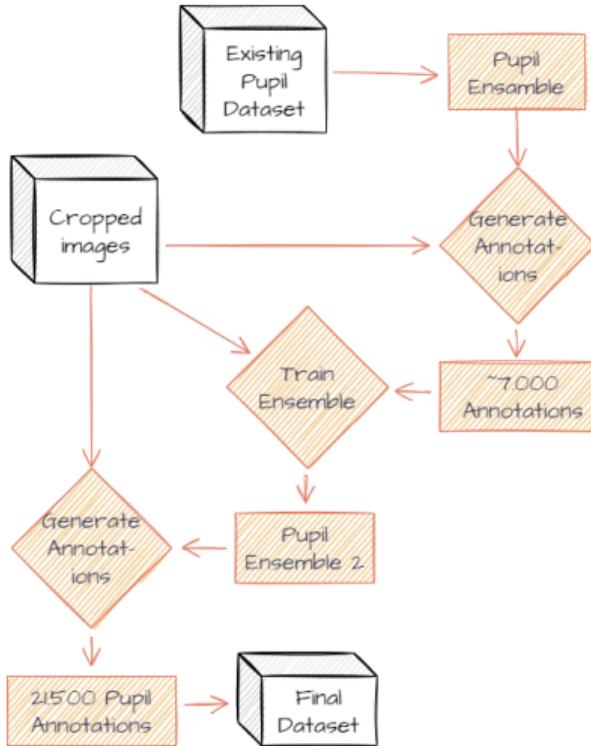
# Unity Eyes



# Datensatz Pipeline



# Datensatz Pipeline



- 1 Training eines Ensembles aus 5 Modellen auf dem alten Datensatz
- 2 Erstellen von Annotationen für die ausgeschnittenen Bilder
- 3 Training eines neuen Ensembles
- 4 Erstellen neuer Annotationen

Ensembles Kriterien:

- Mindestens 2/5 erkennen die Pupille
- Konfidenz von über 70%
- Standardabweichung zwischen Erkennungen muss kleiner als 3 Pixel sein

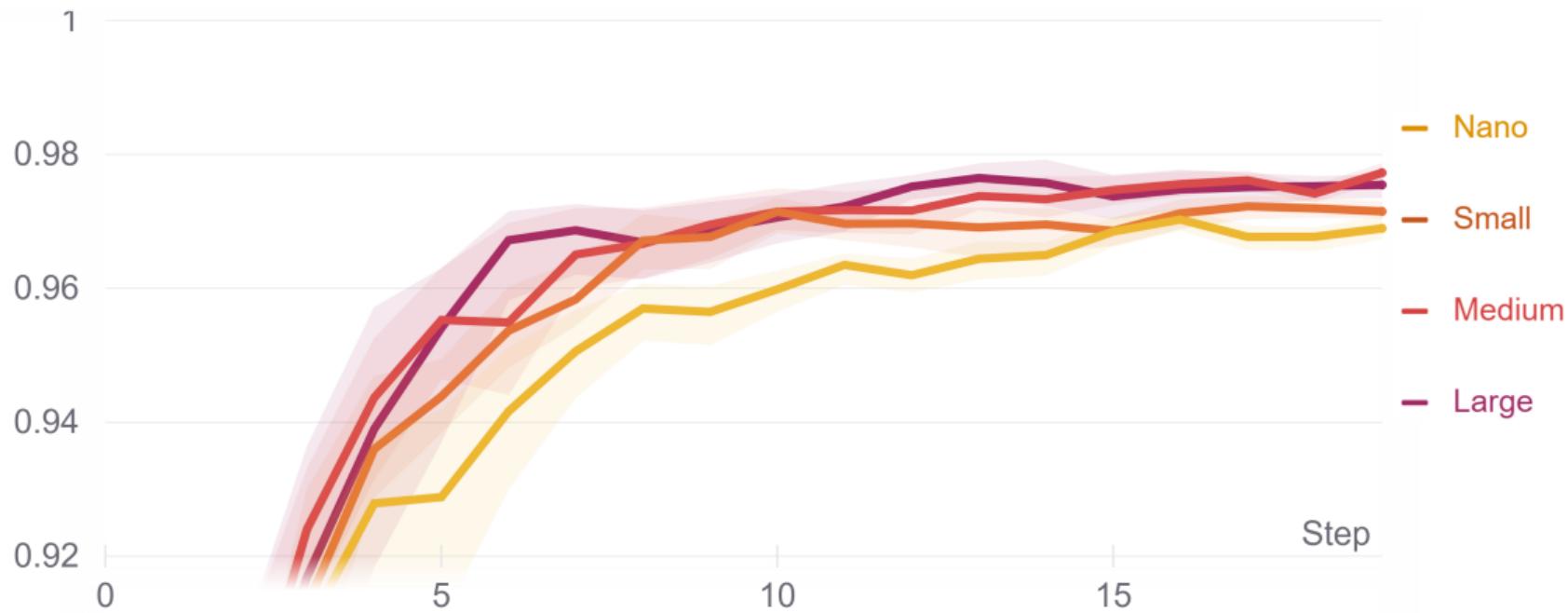
# Experimente

Auswertung der Architekturen und Hyperparameter

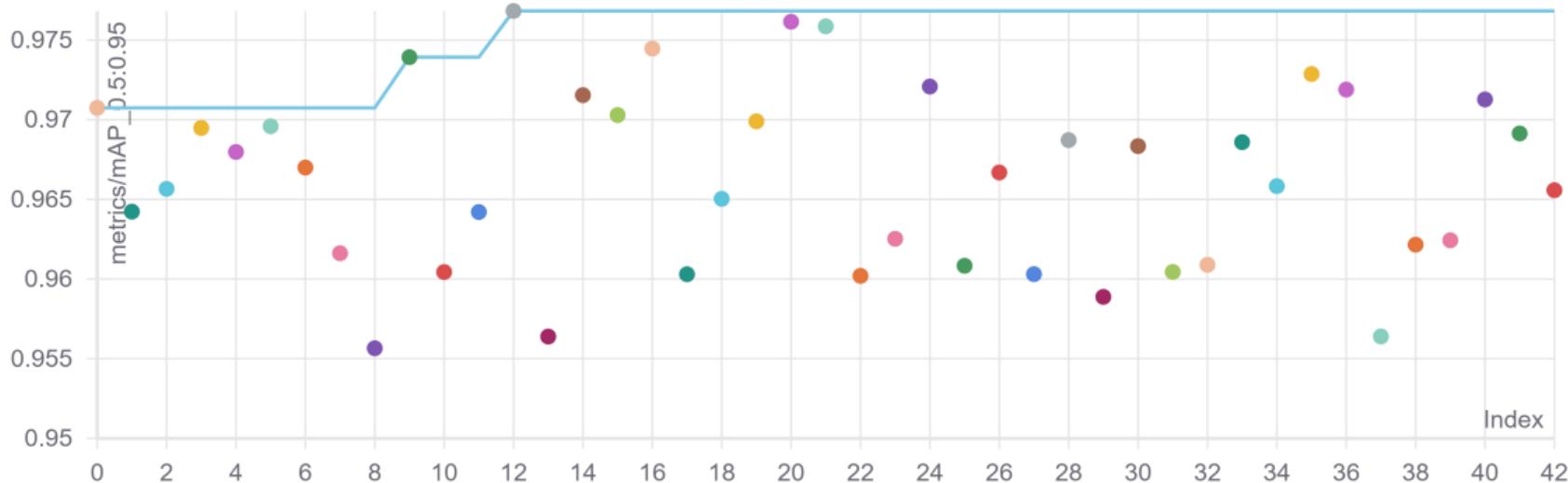
# Vergleich verschiedener YOLO Größen

Model	mAP.95	V100 (ms)	Params (M)	GFLOPs@640
Nano	28.4	6.3	1.9	4.5
Small	37.2	6.4	7.2	16.5
Medium	45.2	8.2	21.2	49.0
Large	48.8	10.1	46.5	109.1
Xtra	50.7	12.1	86.7	205.7

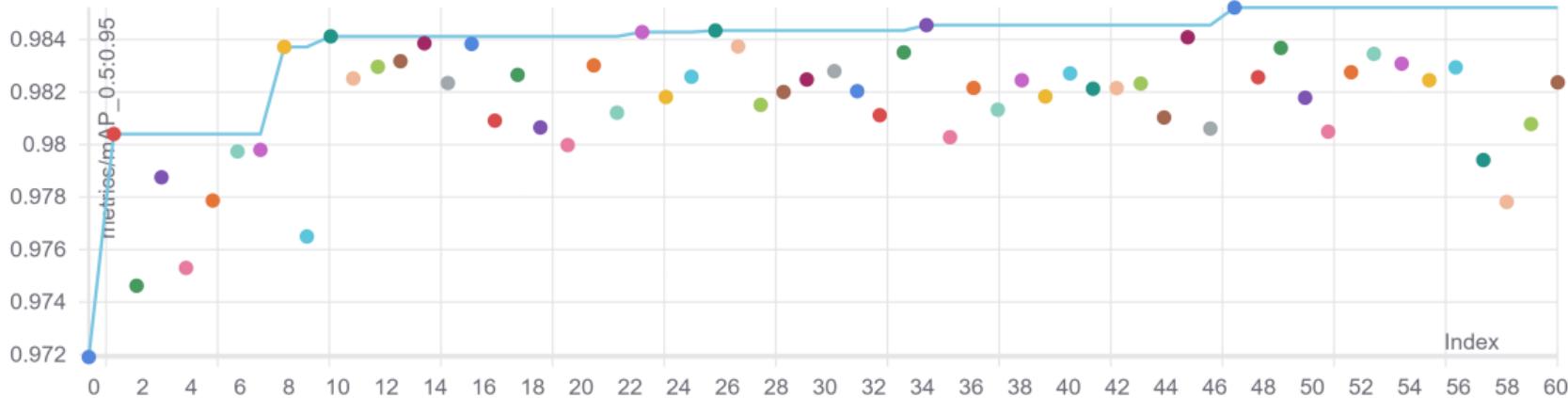
# Vergleich verschiedener YOLO Größen — mAP.95



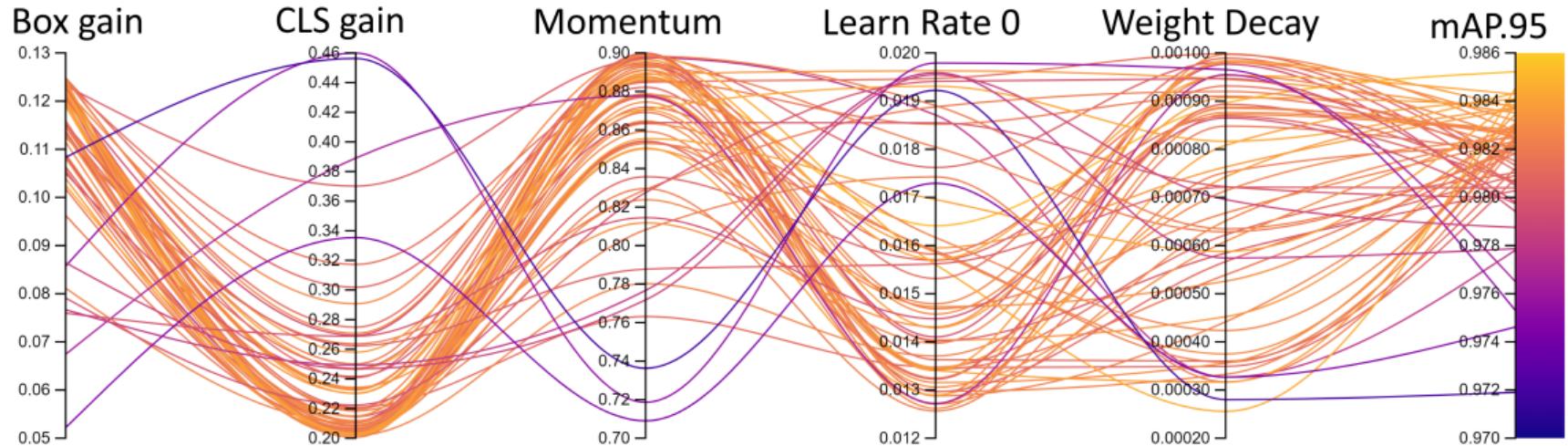
# Zufällige Suche — mAP.95 Optimierung



# Bayes'sche Suche — mAP.95 Optimierung



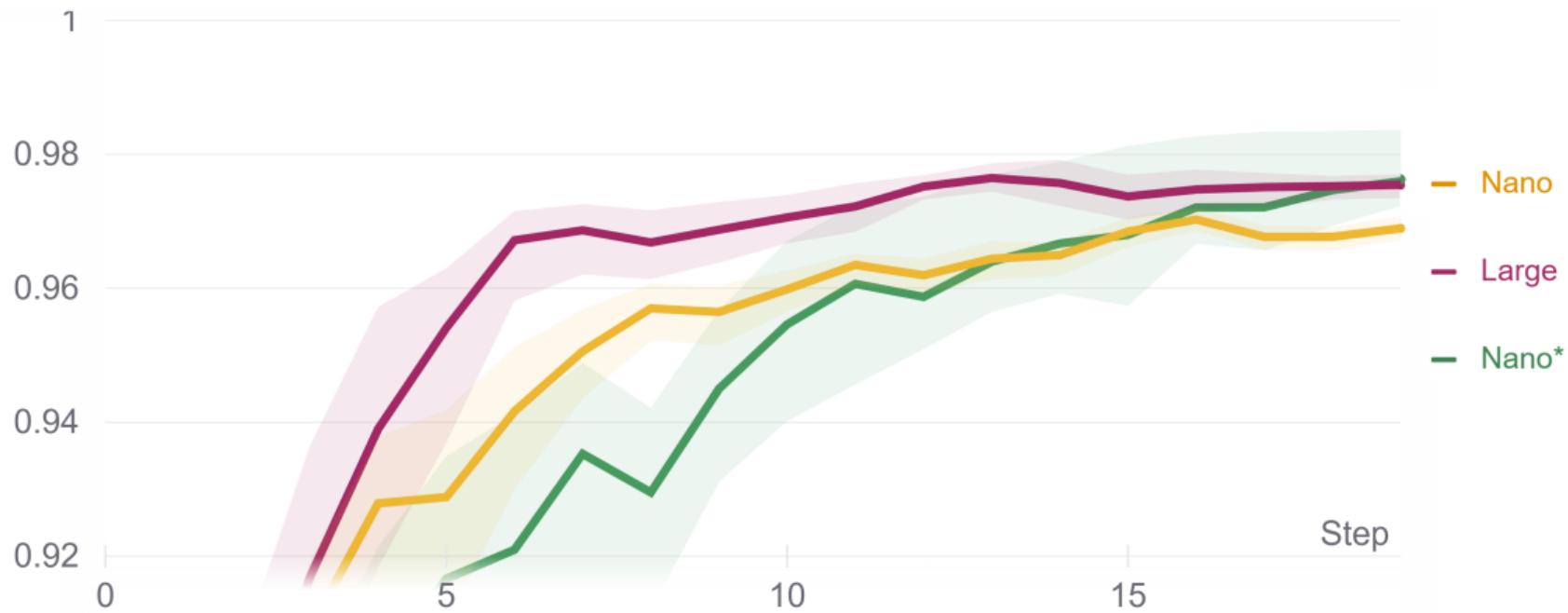
# Bayes'sche Suche — mAP.95 Optimierung



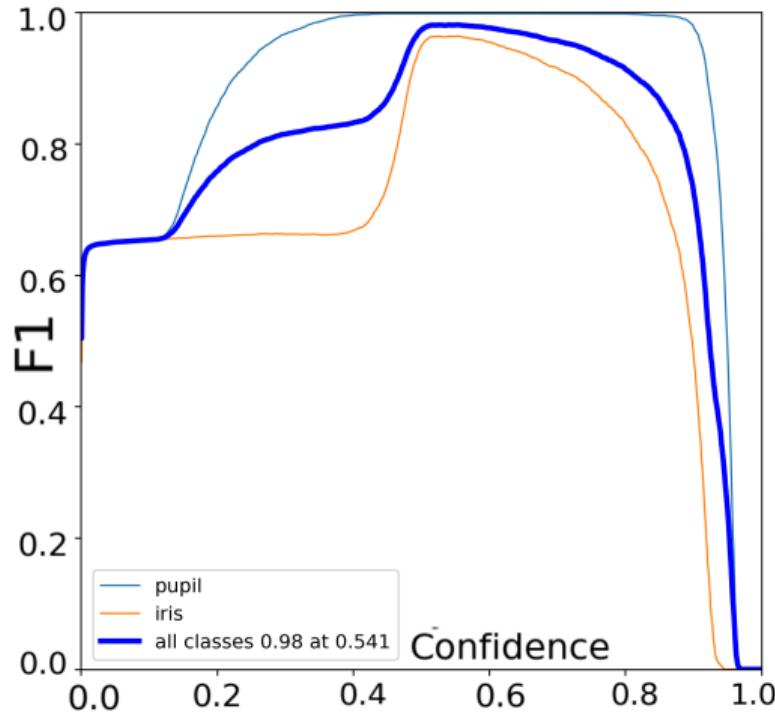
# Bayes'sche Suche — mAP.95 Optimierung

Training	Stärke	Korrelation	Eingabe	Stärke	Korrelation
Class Loss Gain	0.442	-0.724	Scale	0.200	0.500
Box Loss	0.068	0.352	Color Saturation	0.038	0.142
Momentum	0.043	0.165	Batch Size	0.035	0.181
Learning Rate 0	0.042	-0.317	FlipLR	0.031	0.048
Weight Decay	0.030	-0.025	Color Value	0.026	0.156
			Color Hue	0.025	0.013
			Translate	0.019	-0.017

# Nano mit optimalen Hyperparametern — mAP.95



# Schwellwertbestimmung

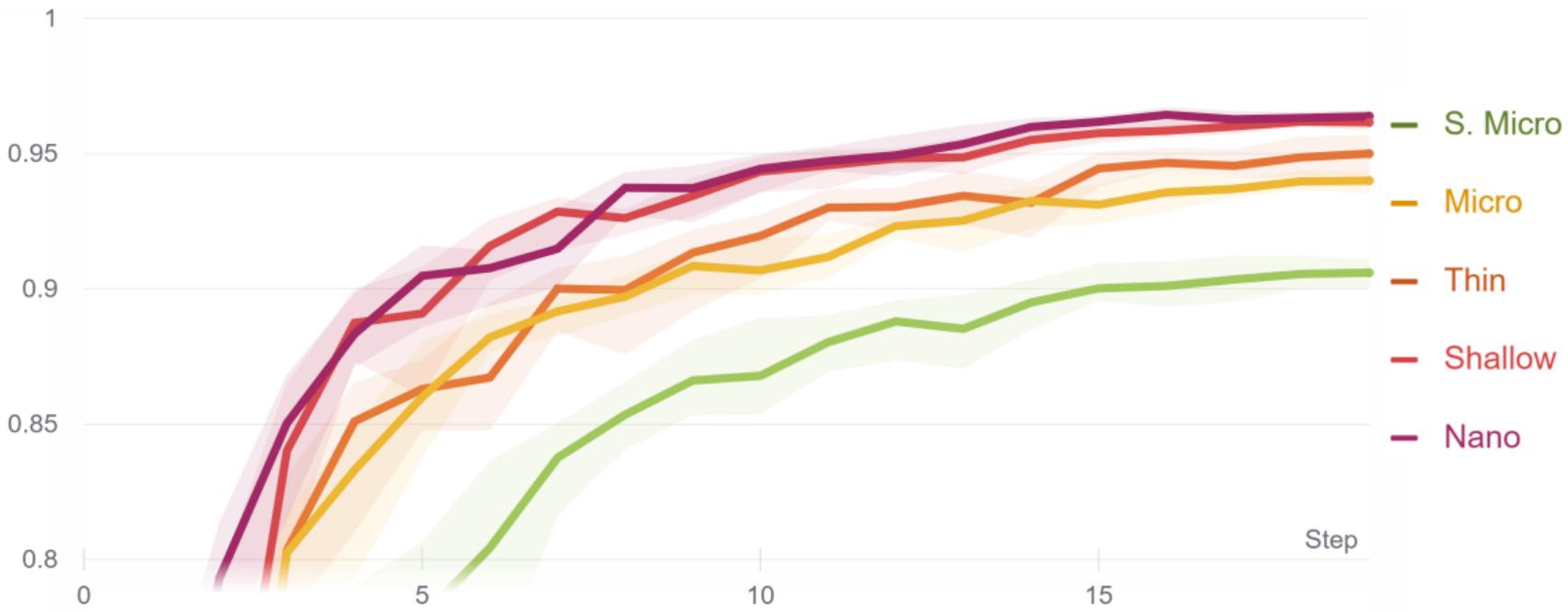


- Zusammengesetzt aus Precision und Recall
- Steigt der Schwellwert sinkt der Recall
- Sinkt der Schwellwert steigt die Precision
- Optimaler Wert für  $F_1$ : 0.541
- Größerer Wert für besserer Genauigkeit

# Schrumpfen des Modells

Model	Depth Scaling	Width Scaling	Layers	Parameters	GFLOPs
Large	1.0	1.0	335	46.000.000	109.1
Nano	0.33	0.25	270	1.766.623	4.2
Shallow	0.125	0.25	243	1.673.823	3.8
Thin	0.33	0.1	270	329.207	1.0
Micro	0.125	0.1	243	310.679	0.9
Super Micro	0.05	0.05	192	94.651	0.4

# Schrumpfen des Modells — mAP.95





# Schluss

Android, Fazit und Ausblick

# Android Implementation

- Übersetzen in Kotlin:
  - Komfortfunktionen wie Null-Pointer Safety oder Scoped Functions
  - View Binding für vereinfachten Zugriff auf View-Properties
- Kamera API:
  - Alte API stellt nur fundamentale Features zur Verfügung
  - CameraX kommt mit ImageAnalysis und weiteren AI-Features
- PyTorch Mobile:
  - Wechsel von TFLite und dem SSDLite MobileNetV3 zu YOLOv5
  - Probleme mit Kompatibilitäten

# Fazit

- Vorstellung des Unity Eyes Datensatzes für die Erkennung von Iris und Pupille
- Untersuchung unterschiedlicher Hyperparameter und Architekturen
  - ▣ YOLOv5 Größen, eigenes Verkleinern
  - ▣ Eingabeaugmentationen, Lernparameter
- Durch Optimierung kann Nano zu Large aufschließen
- Objekterkennung in Echtzeit ist möglich
- Sehr gute Erkennungsraten und Genauigkeiten

# Ausblick

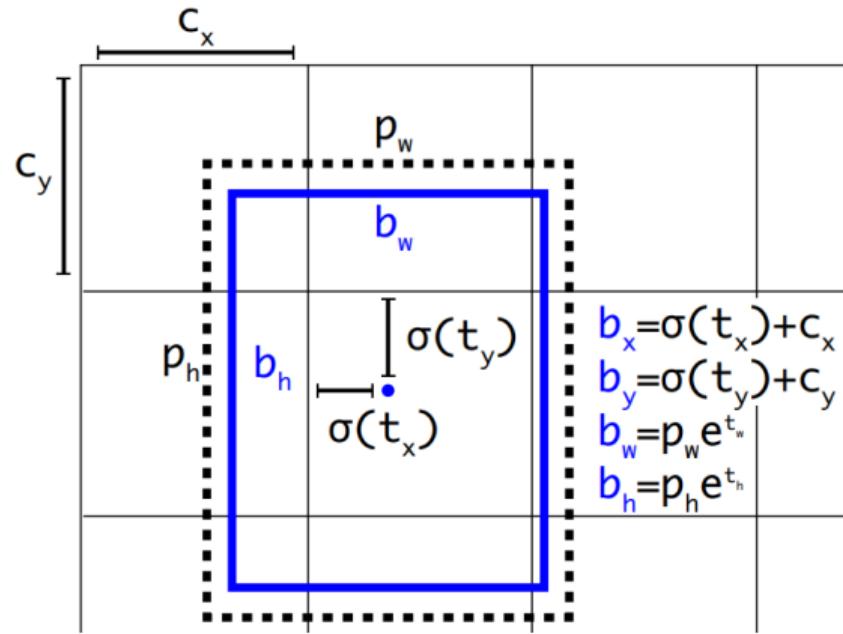
- Deep Learning Modell
  - ▣ Einfluss von Pruning und Quantisierung
  - ▣ Andere Architekturen?
- Android App:
  - ▣ Analyse-Funktion, Auswertung und Visualisierung
  - ▣ Studie zum Einsatz außerhalb von Laborbedingungen
- Einsatz in der Medizin
  - ▣ Soziale Verträglichkeit
  - ▣ Überprüfbarkeit



# Appendix

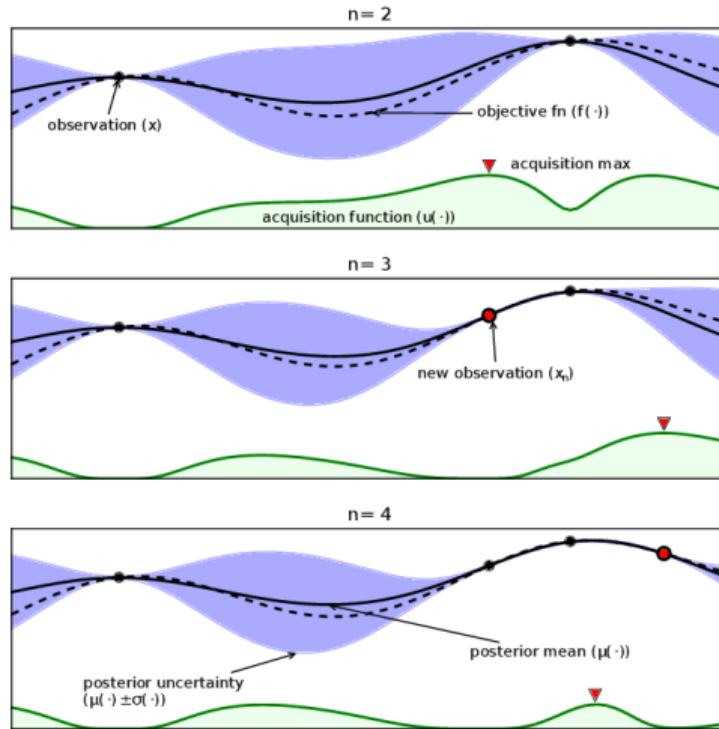
YOLO, Suche, Ergebnisse

# YOLO — Regression

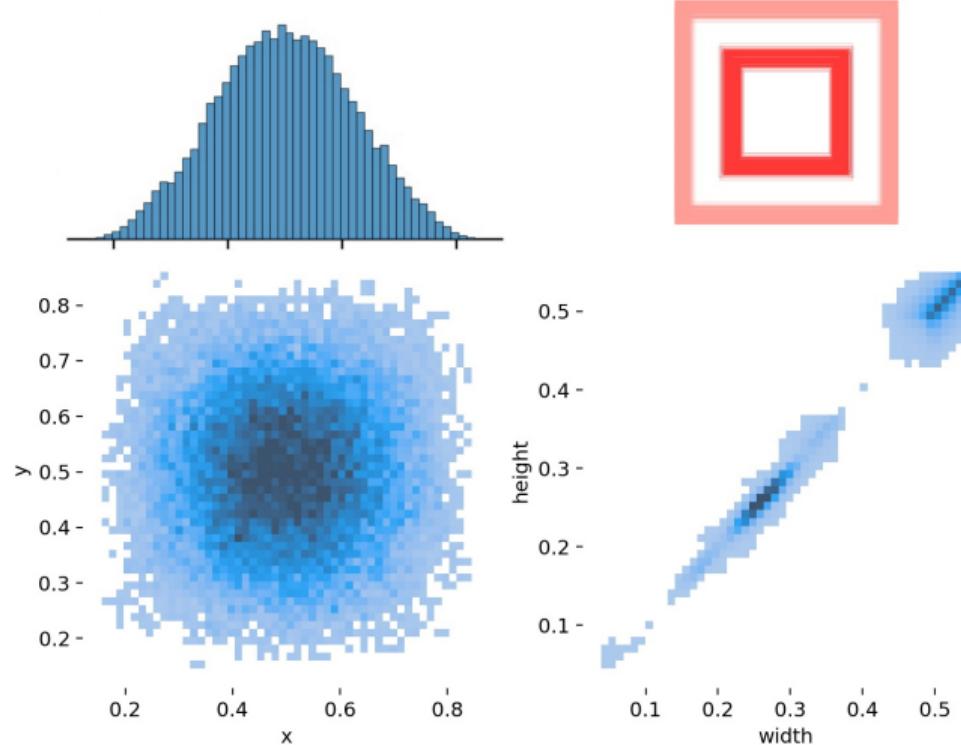


[RF17]

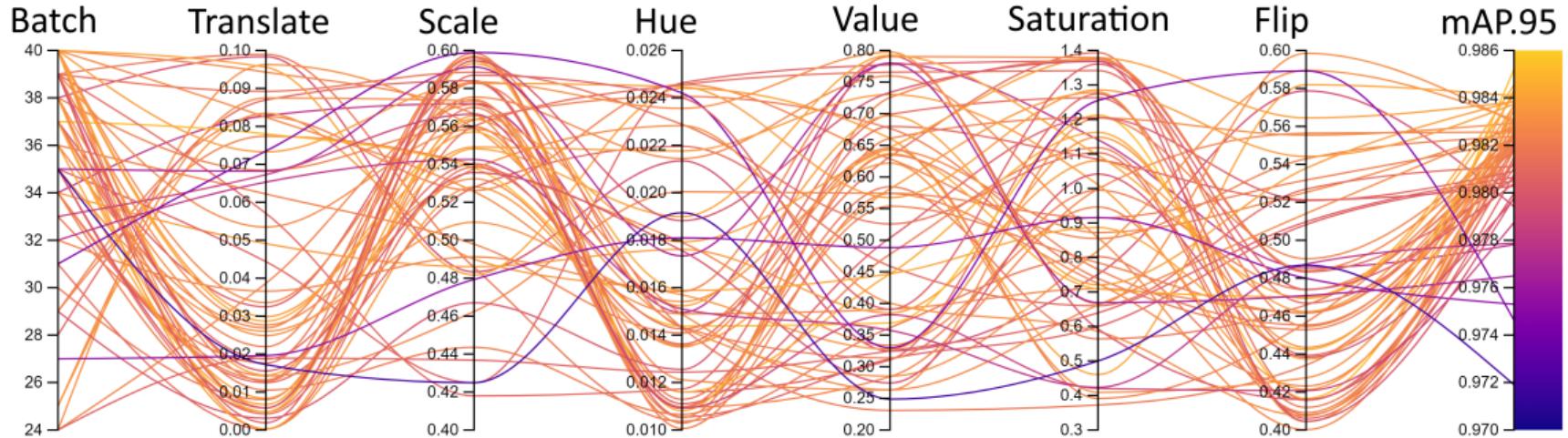
# Bayes'sche Suche



# Unity Eyes Datensatzverteilung



# Bayes'sche Suche — mAP.95 Optimierung



- [Bab18] Sanjana Babu.  
Anchor boxes for object detection.  
<https://www.mathworks.com/help/vision/ug/anchor-boxes-for-object-detection.html>, 2018.  
[Online; accessed 16.11.2021].
- [BB12] James Bergstra and Yoshua Bengio.  
Random search for hyper-parameter optimization.  
*J. Mach. Learn. Res.*, 13:281–305, 2012.
- [Els14] Ines Elsenhans.  
Das schädel-hirn-trauma – wichtige fakten zu pathogenese und therapie.  
<https://www.thieme.de/viamedici/klinik-faecher-orthopaedie-und-unfallchirurgie-1540/a/das-schaedel-hirn-trauma-20464.htm>, 2014.  
[Online; accessed 6.12.2021].
- [Fli] Flickr.  
Flickr eye, licensed under <https://creativecommons.org/licenses/by-nc-sa/4.0/>.  
<https://www.flickr.com/>.  
[Online; accessed 9.09.2021].
- [JSB<sup>+</sup>20] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, Ayush Chaurasia, TaoXie, Liu Changyu, Abhiram V, Laughing, tkianai, yxNONG, Adam Hogan, lorenzomammanna, AlexWang1900, Jan Hajek, Laurentiu Diaconu, Marc, Yonghye Kwon, oleg, wanghaoyang0106, Yann Defretin, Aditya Lohia, ml5ah, Ben Milanko, Benjamin Fineran, Daniel Khromov, Ding Yiwei, Doug, Durgesh, and Francisco Ingham.  
yolov5.  
<https://github.com/ultralytics/yolov5>, 2020.  
commit cd35a009ba964331abccd30f6fa0614224105d39.

- [Kim17] Sung Hee Kim.  
Approach to pupillary abnormalities via anatomical pathways.  
*Yeungnam University Journal of Medicine*, 34:11–18, 2017.
- [LDS89] Yann LeCun, John S. Denker, and Sara A. Solla.  
Optimal brain damage.  
In *NIPS*, 1989.
- [LJD<sup>+</sup>17] Lisha Li, Kevin G. Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet S. Talwalkar.  
Hyperband: A novel bandit-based approach to hyperparameter optimization.  
*J. Mach. Learn. Res.*, 18:185:1–185:52, 2017.
- [Mor21] Alexander Moryäner.  
Entwicklung einer app zur diagnoseunterstützung durch analyse der pupillengröße per smartphonekamera, 1 2021.
- [Pan19] Ayush Pant.  
Introduction to machine learning for beginners.  
<https://towardsdatascience.com/introduction-to-machine-learning-for-beginners-eed6024fdb08>, 2019.  
[Online; accessed 6.12.2021].
- [RF17] Joseph Redmon and Ali Farhadi.  
Yolo9000: Better, faster, stronger.  
*2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017.
- [Rot11] Peter M. Roth.  
On-line conservative learning.  
04 2011.
- [SSW<sup>+</sup>16] Bobak Shahriari, Kevin Swersky, Ziyun Wang, Ryan P. Adams, and Nando de Freitas.  
Taking the human out of the loop: A review of bayesian optimization.  
*Proceedings of the IEEE*, 104:148–175, 2016.

- [Wal14] Walber.  
Precision and recall.  
[https://en.wikipedia.org/wiki/Precision\\_and\\_recall#/media/File\protect\leavevmode@ifvmode\kern+.2222em\relaxPrecisionrecall.svg](https://en.wikipedia.org/wiki/Precision_and_recall#/media/File\protect\leavevmode@ifvmode\kern+.2222em\relaxPrecisionrecall.svg),  
2014.  
[Online; accessed 9.10.2021].
- [WBZ<sup>+</sup>15] Erroll Wood, Tadas Baltrusaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling.  
Rendering of eyes for eye-shape registration and gaze estimation.  
In *Proc. of the IEEE International Conference on Computer Vision (ICCV 2015)*, 2015.
- [YOH] SHIVY YOHANANDAN.  
map (mean average precision) might confuse you!  
<https://xailient.com/map-mean-average-precision-might-confuse-you/>.  
[Online; accessed 7.11.2021].