Die Annotation wurde mit dem Editor makesense.ai durchgeführt. Für das Training wurde das Modell YOLOv5 verwendet, das Training wurde im Google colab Notebook durchgeführt (Laufzeittyp GPU).

Insgesamt wurden für das Projektes 10 Sehenswürdigkeiten (Klassen) ausgewählt.

Das Brandenburger Tor (Berlin)
Das Bundestagsgebäude (Berlin)
Der Berliner Fernsehturm
Der Rheinauhafen (Köln, Nordrhein Westfallen)
Die Alpspitze (Bayern)
Der Kölner Dom (Nordrhein Westfallen)
Das Völkerschlachtendenkmal (Leipzig, Sachsen)
Frauenkirche (München, Bayern)
Kreidefelsen (Ostseeküste, Mecklenburg Vorpommern)
Schloss Neuschwanstein (Füssen, Bayern)

Der Annotationsprozess wurde von allen Teammitgliedern unter Vorbehaltung von Annotationsrichtlinien durchgeführt:

- Object Detection
- Rechteckiger Ausschnitt
- Auf den Bildern, auf denen das Objekt teilweise von einem anderen Objekt verdeckt war, oder ein anderes Objekt im Vordergrund zu sehen war, wurde das Objekt komplett annotiert.
- Bei der Annotation des Bundestagsgebäudes wurden die Fahnen auf den Türmen, auf der Kuppel sowie die Fahnen vor dem Zentralen Eingang berücksichtigt.

Jedes Teammitglied hat das Training mit den von ihm annotierten Klassen (Sehenswürdigkeiten) durchgeführt um mögliche Annotationsfehler, fehlende Daten oder sonstige Fehler zu entdecken und zu eliminieren.

Insgesamt wurden 354 Bilder für das Training und 138 für die Validation verwendet.

Die Parameter:
350 Epochen
Batch 16
Optimizer: SGD with parameter groups 57 weight, 60 weight (no decay), 60 bias

Die Anzahl an Bildern (Instanzen) jeder Klasse ist nicht exakt gleich. Da das Modell Yolov5 eine Mehrfachklassifikation durchführt, beeinflusst die Datenverteilung (die Anzahl an Instanzen je Klasse) nicht das Ergebnis.

Das Modell zeigte beim Training mit dem gesamten Datensatz folgende Ergebnisse (siehe Abbildungen 1 – 4.

Abbildung 1 Ergebnisse der Validation (gesamter Datensatz).

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 1
all	138	144	0.877	0.906	0.898	0.604
BrandenburgerTor	138	11	0.882	1	0.995	0.551
Reichstag	138	21	0.921	0.952	0.969	0.74
Fernseherturm	138	21	0.853	0.952	0.949	0.727
Rheinau- hafen	138	24	0.92	1	0.989	0.783
Alpspitze	138	10	0.593	0.585	0.49	0.155
Kölner Dom	138	10	0.942	1	0.995	0.835
Volkerschlachtdenkmal	138	14	0.873	0.929	0.986	0.824
Frauenkirche	138	11	1	0.995	0.995	0.562
Kreidefelsen	138	12	0.79	0.75	0.686	0.301
Schloss Neuschwanstein	138	10	1	0.899	0.927	0.558

Abbildung 2. Confusion-Matrix

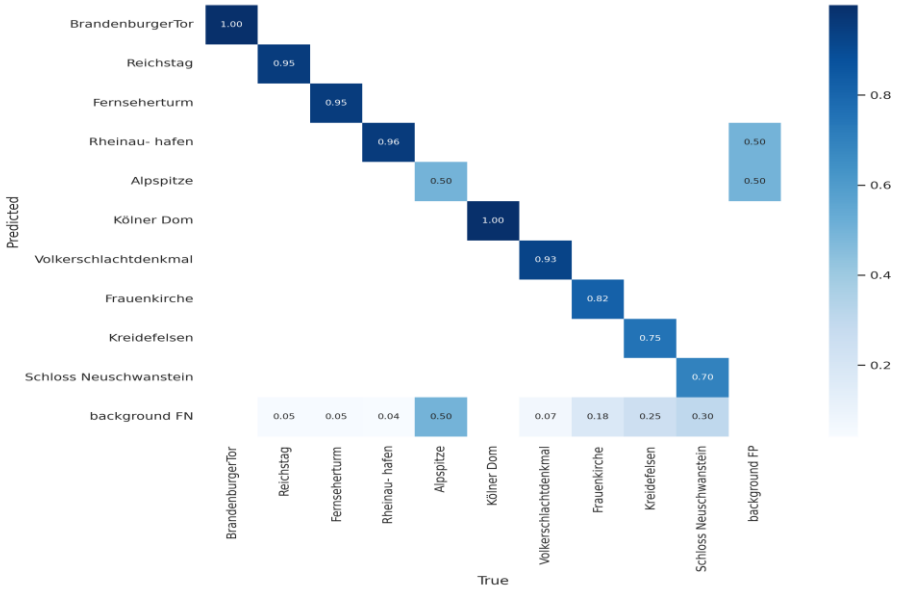


Abbildung 3. Precision-Recall Curve

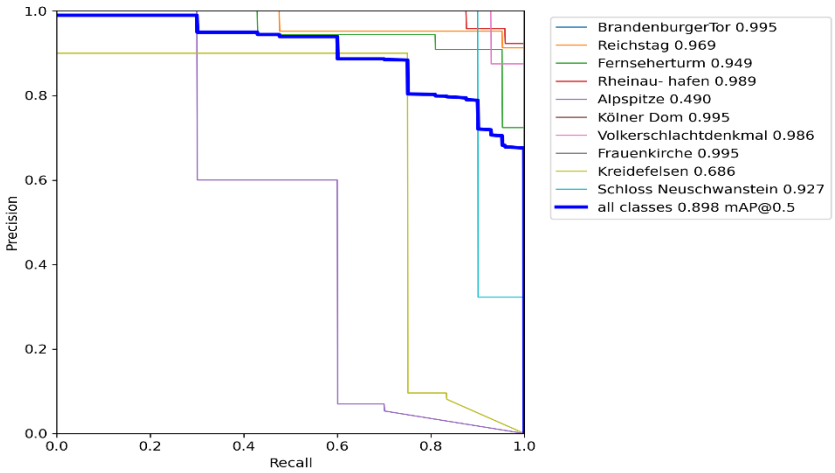
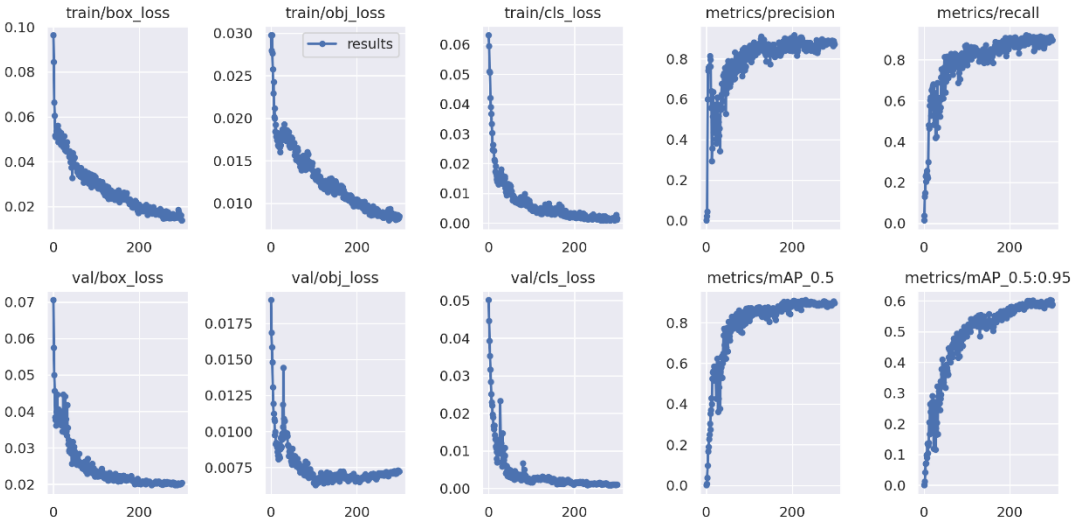


Abbildung 4. Graphische Darstellung des Training/Validation Prozesses



Die Performance-Metriken unterscheiden sich je nach Objekt (Sehenswürdigkeit) sowohl bei einzelnen Trainings-/Validations- Prozessen als auch beim Training mit allen Klassen (gesamter Datensatz).
Je einzigartiger das Objekt, desto einfacher ist es, das Modell zu trainieren, dieses Objekt zu erkennen und von den anderen zu unterscheiden. Dazu kommt noch, dass es schwerer ist, ein ähnliches (falsches) Objekt zu finden, welches das Modell mit der wahren Klasse verwechseln könnte. Dadurch bekommt man bei solch einzigartigen Objekten bessere Ergebnisse.
Umgekehrt ist es bei den sogenannten natürlichen Sehenswürdigkeiten, vor allem für die „Alpspitze“. Die Scores für dieses Objekt sind am niedrigsten, da das Modell Berge voneinander kaum unterscheiden kann und fast jede Gebirgslandschaft als „Alpspitze“ klassifiziert.

Die Abbildungen 5 und 6 zeigen, dass sich die Precisionkurven für die „Alpstitze“ und die „Kreidefelsen“ von den anderen wesentlich zum negativen unterscheiden, beim Recall Score zeigte das Modell für die „Alpspitze“, „Schloss Neuschwanstein“ und die „Kreidefelsen“ schlechtere Ergebnisse als für die anderen.

Abbildung 5. Recall Kurve (Abhängigkeit von der Konfidenz)

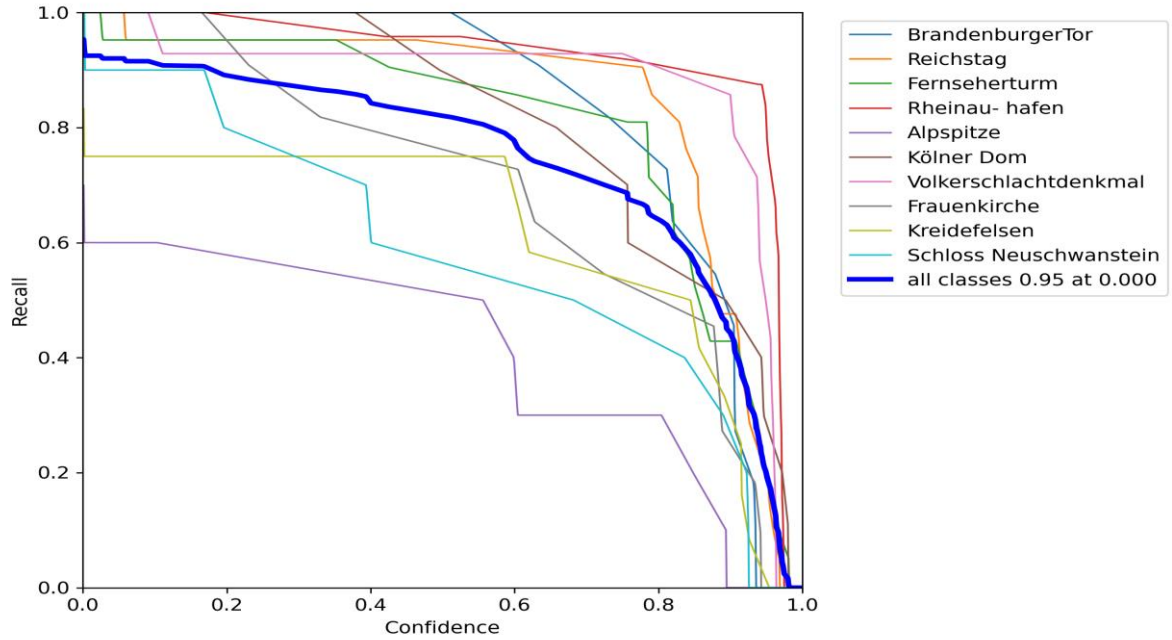
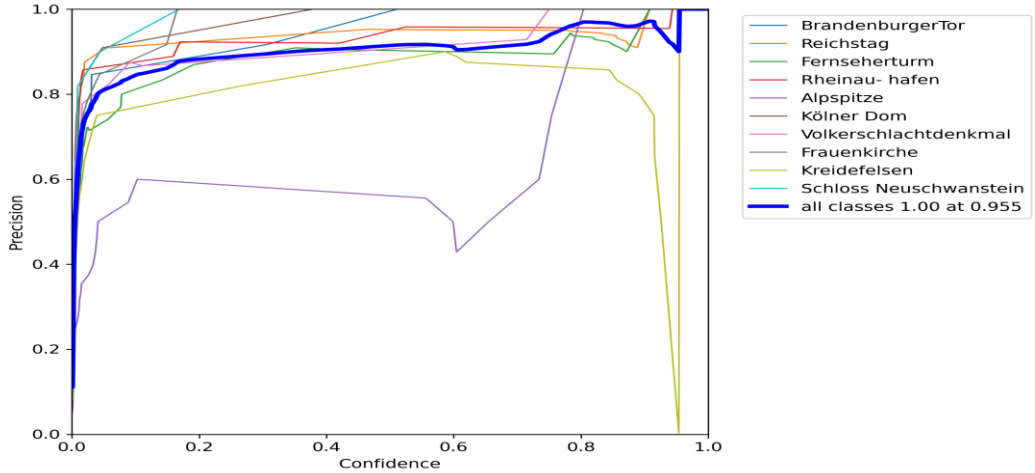


Abbildung 6. Precision (Abhängigkeit von der Konfidenz)



Der zweite Faktor, welcher beim Training und bei den Evaluierungsergebnissen eine wichtige Rolle spielt sind die Trainingsdaten selbst, beziehungsweise die Verteilung der „falschen“ Daten in Trainings/Validationen und Test. Die Scores ändern sich je nachdem, wie die Erweiterung des Datensatzes erfolgt. Ein ausführlicheres Beispiel dazu gibt es im nächsten Abschnitt (den Trainingsprozess des Modells mit den Sehenswürdigkeiten Berliner Fernsehturm, Bundestagsgebäude).

Interessant war auch zu beobachten, dass die Ergebnisse in den Testphasen nach den Einzeltrainingsverfahren und nach dem Training mit den gesamten Daten einen wesentlichen Unterschied gezeigt haben.

Nach dem ersten Training mit 200 Epochen sind viele Instanzen aus der false negativ Kategorie in die true positive gewandert. Gleichzeitig entstanden auch einige false positive Klassifikationen, welche in einzelnen Trainingsphasen nicht aufgetreten sind. Zum Beispiel, wurden die „falschen“ Bilder, die für die Klasse „Bundestag“ gewählt wurden nicht als Bundestag sondern als „Brandenburger Tor“ klassifiziert.

Nach dem Training mit 350 Epochen wurden die Testergebnisse besser. Selbst ständig auftauchende Klassifikationsfehler sind in der Trainingsphase nicht aufgetaucht. Das spricht für eine Verbesserung des Modells, jedoch nicht dafür, dass bei weiterer Verwendung, mit weiteren Testdaten keine gleichen Fehler auftreten können. Die Fehlerverteilung mit längerer Beobachtung könnte daher anders sein als in der durchgeführten Testphase.

Bei den unten zu betrachtenden Beispielen haben sich die Performance-Ergebnisse der Testphase nach dem Training mit dem gesamten Datensatz im Vergleich zu der Testphase im Einzeltrainingsverfahren auch deutlich verbessert. (siehe Tabelle 1 und 2). Teilweise verdeckte, rotierte Objekte wurden erkannt und richtig klassifiziert, auf den Bildern mit mehreren Objekten wurden die richtigen Objekte erkannt und den richtigen Klassen zugeordnet.
Der „Raketen-Effekt“ (ausführlicher im unteren Abschnitt beschrieben) tauchte in der Testphase nach dem Training mit dem gesamten Dataset mit 350 Epochen nicht mehr auf.

Der Trainingsprozess des Modells mit den Sehenswürdigkeiten (Klassen) Berliner Fernsehturm, Bundestagsgebäude (als Beispiel):

Datenaufbereitung:
Für das Model wurden Bilder vom heutigen Zustand und Aussehen der Objekte gewählt. Betrachtet wurde nur das äußerliche Aussehen und keine Innenräume. Den Kern des Datensatzes machen die Bilder aus, auf denen die zu erkennenden Objekte komplett zu sehen sind, es keine Störungen gibt. (ca. 60% der Trainingsbilder)
Der Datensatz wurde durch die Data Augmetation Methode erweitert.

- Rotation (bei Fernsehturm)
- Translation/Cropping
- Flipping
- Scaling
- Helligkeit / Kontrast / Schwarz-weiße Bilder
- Unscharfe Bilder
- Unterschiedliche Perspektiven und Winkel
- Bilder, auf denen das Objekt teilweise verdeckt ist

Außerdem wurde der Datensatz durch „falsche“ nicht annotierte Bilder erweitert. Gewählt wurden Gebäude und Objekte, welche von der Bauart, der Konstruktion und der Form ähnlich zu den betrachtenden Sehenswürdigkeiten sind.

Für das Bundestagsgebäude wurden folgende „falsche“ Bilder gewählt (die ähnlichen Gebäude enthalten Säulen, Steinreliefs, eine Kuppel, einen Giebel mit Inschrift):

- Das Bundesratsgebäude (Berlin)
- Das Parlamentsgebäude (Wien)
- Das Parlamentsgebäude (Madrid)
- Das Parlamentsgebäude (Tbilisi)
- Das Bolschoi Theater, Moskau
- Das Puschkin Museum, Moskau
- Der Petersdom, Vatikan

Für den Berliner Fernsehturm wurden folgende „falsche“ Bilder gewählt (ähnliche Fernsehtürme sowie Objekte mit einer ähnlich spitzen Form):

- Der Fernsehturm, Moskau (Russland)
- Der Fernsehturm, Köln (Russland)
- Der Fernsehturm, Düsseldorf (Russland)
- Der Fernsehturm, Baku (Azerbaijan)
- Der Fernsehturm, Toronto (Canada)
- Ein Raketenstart vom kasachischen Weltraumbahnhof Baikonur

Insgesamt wurden 5 Durchläufe gemacht. Die Anzahl an Bildern sowie die Anzahl an Epochen wurden mit jedem Durchlauf erhöht. Die Evaluierungsergebnisse in der Testphase waren niedriger als bei dem Training/Validation (in der Dev Phase), da in den ersten Durchläufen keine „falschen“ Bilder in der Validationsphase vorhanden waren. Deswegen waren die Ergebnisse des Trainings/Validation nicht repräsentativ und zeigten höhere Ergebnisse.

Das Modell zeigt allgemein mehr false positive Klassifikationen für den Fernsehturm als für den Bundestag – d.h. das Modell klassifiziert andere Fernsehtürme als den Fernsehturm. Der Grund dafür ist, dass es mehr ähnliche Fernsehtürme als dem Bundestag ähnliche Gebäude gibt. Außerdem hat das Modell das Bild eines Raketenstarts als den Fernsehturm klassifiziert. Das Problem trat gleich beim ersten Durchlauf auf. Danach wurde die Anzahl an Trainingsdaten allgemein stufenweise erhöht, unter anderem die Anzahl an „falschen“ Bildern. Die „falschen“ Bilder wurden im Training/Validation sowie in der Testphase im fünften Durchlauf hinzugefügt.

Der „Raketen-Effekt“ wurde dadurch nicht eliminiert, obwohl das Model die tatsächlichen Klassen mit jedem weiteren Durchlauf besser erkannt hat.

Schwierigkeiten gab es unter anderem auch mit Nahaufnahmen für beide Klassen (vermehrt beim Bundestag), mit Bildern, wo die Objekte teilweise verdeckt waren (mehr beim Bundestag), rotierte Bilder (mehr beim Fernsehturm), Fernaufnahmen, wo auch viele andere Objekte vorhanden waren (mehr beim Fernsehturm).

Die genauen Zahlen in 5 Durchläufen wurden dokumentiert und die Precision, Recall und Accuracy für die Testphase berechnet. (siehe Tabelle 1).

Tabelle 1 Trainingsverfahren mit den Klassen „Bundestag“, „Fernsehturm“

	Erster Durchlauf		Zweiter Durchlauf		Dritter Durchlauf		Vierter Durchlauf		Fünfter Durchlauf	
	Bundestag	Fernsehturm	Bundestag	Fernsehturm	Bundestag	Fernsehturm	Bundestag	Fernsehturm	Bundestag	Fernsehturm
Anzahl an Bildern für Training	30	30	40	40	70	70	70	70	82	82
davon: Anzahl an "falschen" nicht annotierten Bildern	5	5	5	5	7	20	7	20	7	20
Anzahl an Bildern für Val	10	10	15	15	21	21	21	21	26	26
davon: Anzahl an "falschen" nicht annotierten Bildern	0	0	0	0	0	0	0	0	5	5
Test	10	10	13	13	23	23	23	23	23	23
Epochen	100		200		250		300		300	
mAP	**0.797**	**0,968**	**0.973**	**0.987**	**0.981**	**0.974**			**0,955**	**0,956**
Precision (Testphase)	wurde nicht berechnet, da mAP beim Val(Dev) sowieso ziehmlich niedrich war		**1,00**	**0,64**	**1,00**	**0,67**	fast keine Änderung		**1,00**	**0,67**
Recall (Testphase)			**0,80**	**0,78**	**0,68**	**0,56**			**0,68**	**0,56**
Accuracy (Testphase)			**0,85**	**0,54**	**0,74**	**0,43**			**0,74**	**0,43**
True positive			8	7	13	10			13	10
False positive			0	4	0	5			0	5
False negative			2	2	6	8			6	8
True negative			3	0	4	0			4	0

Tabelle 2. Die Performance Metriken der betrachtenden Klassen in der Testphase nach dem Training des Modells mit dem gesamten Datensatz

	Bundestag	**Fernsehturm**
True positive	20	20
False positive	2	1
False negative	1	2
Precision	**0,90**	**0,95**
Recall	**0,95**	**0,90**

Der „Raketen-Effekt“ und die Precision- und Recall- Scores in der Testphase zeigten, dass es ein Verbesserungspotential für das Model gibt. Der Raketten-Effekt kam jedoch in der Testphase nach dem Training mit dem gesamten Dataset nicht mehr vor. Das Modell kann Fernsehtürme voneinander immer noch nicht zu 100 Prozent unterscheiden. Es ist nicht das Ziel des Projektes, dass das Modell alle Fernsehtürme voneinander unterscheidet.

Das Training mit mehr als zwei Klassen führte allgemein zu besseren Ergebnissen als das Einzeltraining. Das Modell wurde robuster, da es die Features von mehreren Klassen „gelernt“ hat, und dadurch wurde die allgemeine Fehlerquote gesenkt.