

# Trainieren eines KI-Modells zur Objekterkennung mit einem benutzerdefinierten Datensatz



Whitepaper

**KEBA**<sup>®</sup>  
Automation by innovation.

# Trainieren eines Objekts

---

Das erneute Trainieren eines bereits trainierten Modells mit einem benutzerdefinierten Datensatz ist bei der visuellen KI-Verarbeitung entscheidend. Nur so ist das Modell in der Lage, sich an die konkreten Eigenschaften, Muster und Merkmale der Zielanwendung anzupassen. Während vorab trainierte Modelle eine gute Grundlage bilden, indem sie allgemeine Merkmale von großen Datensätzen lernen, ist das Finetuning mit domänenspezifischen Daten unerlässlich, um eine höhere Genauigkeit, Robustheit und Relevanz für die jeweiligen Aufgaben zu gewährleisten.

In unserem Beispiel trainieren wir ein KI-Modell zur Objekterkennung, mit dem Pralinen auf einem Förderband erkannt werden sollen. Wie beim kindlichen Lernprozess müssen auch intelligente Anlagen lernen, welche Objekte in einem Bild vorhanden sind und wie sie voneinander unterschieden werden können. In diesem Fall bedeutet das, dass der Trainingsalgorithmus des KI-Modells mit einem großen Satz an Bildern gespeist wird, der die zu erkennenden Objekte enthält. Anhand dieser Bilder lernt das Modell, die Objekte korrekt vom Hintergrund und von anderen Objekten zu unterscheiden. Das Modell wird dann anhand dieser erkannten Objekte erstellt und in der KI-Laufzeitanwendung eingesetzt. Damit unsere Anlage beispielsweise das Erkennen von Pralinen erlernen kann, benötigt sie eine große Anzahl von Bildern der Pralinen aus verschiedenen Winkeln, vor verschiedenen Hintergründen, bei unterschiedlichen Lichtverhältnissen und in verschiedenen Positionen, Farben und Details. Nur dann kann sie erkennen, welche Merkmale die Pralinen von anderen Objekten unterscheiden. Die für das Training benötigte Anzahl der Bilder hängt immer vom Anwendungsfall und davon ab, wie einheitlich die Umgebungsbedingungen in der letztendlichen Anwendung sind, z. B. Licht oder Variationen der Objekte. In unserem Fall gehen wir von recht stabilen Lichtverhältnissen aus, sodass ca. 500 Bilder von Pralinen ausreichen sollten, um hinreichend gute Ergebnisse für den vorliegenden Anwendungsfall zu erzielen. Große Modelle mit vielen verschiedenen Objekten, die in den unterschiedlichsten Umgebungen, darunter auch im Freien usw., eingesetzt werden sollen, werden jedoch häufig mit Zehntausenden von Bildern trainiert.

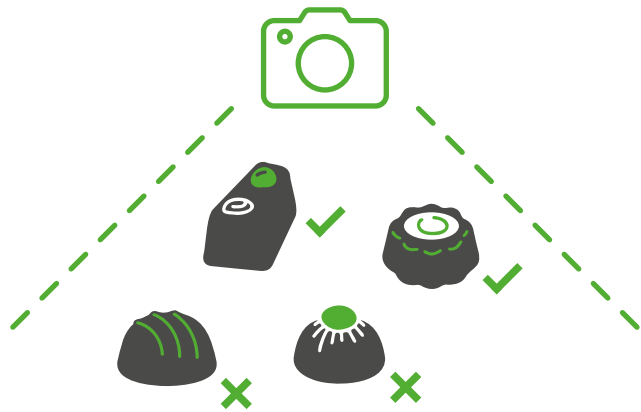
Der Trainingsprozess ist in der Regel der zeitaufwändigste Teil bei der Entwicklung einer KI-Anwendung. Dies liegt vor allem an der Größe der Datensätze. Bei größeren Datensätzen ist der zeitliche Aufwand für das Trainieren in der Regel höher, da das Modell mehr Beispiele verarbeiten und daraus lernen muss. Die Trainingsstrategie hat ebenfalls großen Einfluss. Dazu gehören die Wahl des Optimierungsalgorithmus, die Lerngeschwindigkeit, die Regularisierung und die für das Training verwendete Hardware. Für ein optimales Ergebnis kann es notwendig sein, mit verschiedenen Strategien zu experimentieren (<https://encord.com/blog/fine-tuning-models-hyperparameter-optimization/>).

In unserem Beispiel basiert das Training auf dem überwachten Lernen. Dieser Ansatz ist eine der drei am häufigsten verwendeten Methoden für das maschinelle Lernen. Er nutzt markierte bzw. gelabelte Beispiele für die gesuchten Objekte, d. h. diese sind die „richtigen“ Antworten beim Vergleich mit anderen Bildern. Der Prozess ist folgender:

- Das Modell für maschinelles Lernen wird mit einem Satz gelabelter Trainingsdaten trainiert.
- Anschließend trifft es Annahmen anhand Labels der nachfolgenden Testdatensätze, oft mit mehreren Wiederholungen, bis...
- es für den Einsatz mit Daten der echten Anwendung bereit ist.

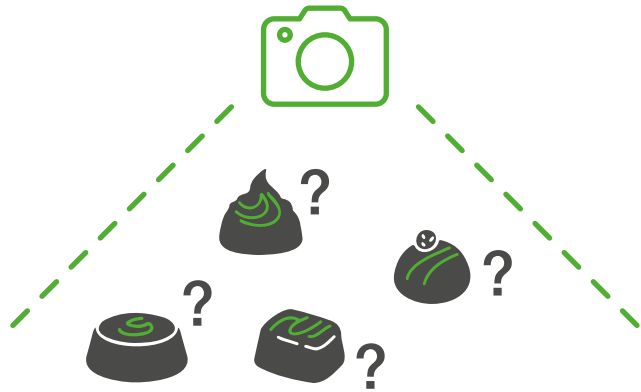
### Überwachtes Lernen:

In einem ersten Schritt werden die Labels durch einen Menschen vergeben (= Trainingsdatensatz).



### Zweiter Schritt:

Die KI erhält einen echten Datensatz und muss selbständig entscheiden.



Das Trainieren eines tiefen neuronalen Netzes mit einem benutzerdefinierten Datensatz für die visuelle Verarbeitung in einem industriellen Kontext kann von einigen Stunden bis zu mehreren Tagen dauern. In unserem Fall dauerte die Erstellung einer ersten Version des Trainingsdatensatzes etwa acht Stunden (Erstellen und Labeln der Bilder der Pralinen). Das Training des Modells mit dem Algorithmus selbst dauerte etwa drei Stunden. Es waren jedoch drei Wiederholungen und eine Anpassung des ersten Datensatzes notwendig, um ein Modell zu erhalten, das unseren Anforderungen entsprach.

# Schritt für Schritt: Wie wird ein Modell trainiert?

- 1** Einen Trainingsdatensatz erstellen
- 2** Das Modell trainieren und testen
- 3** Das Modell in einer echten Anwendung verwenden



# 1. Einen Trainingsdatensatz erstellen

Um ein Modell mittels überwachtem Lernen zu trainieren, muss zunächst ein Trainingsdatensatz erstellt werden. Die Qualität des Modells hängt hauptsächlich von diesen Daten ab. Sind die Daten nicht genau genug oder zu homogen, funktioniert das Modell in der echten Anwendung eventuell nicht. Das heißt, die richtigen Daten und die Menge der Daten sind ausschlaggebend. Der Trainingsdatensatz sollte die Informationen enthalten, die das Modell später lernen soll. In unserem Fall möchten wir ein Modell zur Objekterkennung trainieren, das – wie der Name schon sagt – später Objekte in einem Bild erkennen soll. Wir benötigen dafür also einen Trainingsdatensatz mit Bildern der Objekte, die erkannt werden sollen. Die Erstellung eines guten Datensatzes ist nicht einfach. Wir haben wie gesagt etwa 500 Bilder für unser Training verwendet. In unserem Fall sind die Umgebungsbedingungen relativ stabil: Der Hintergrund bleibt gleich, nur das Licht kann sich ändern. Wir haben Bilder bei unterschiedlichen Lichtverhältnissen aufgenommen, d. h. bei Tageslicht und künstlicher Beleuchtung. Wenn andere Objekte, z. B. Hände von Mitarbeiter:innen oder falsche Produkte, von der Kamera erfasst werden könnten, müssen auch Bilder dieser Objekte in den Trainingsdatensatz aufgenommen werden. Da diese Objekte jedoch nicht hervorgehoben werden, lernt das Modell, dass diese Objekte nicht relevant sind. So kann vermieden werden, dass falsche Objekte erkannt werden. Falsch erkannte Objekte können bei KI-Anwendungen immer zu Problemen führen. Hat ein Modell ein Objekt nicht gelernt (es wurde in den Bildern im Trainingsdatensatz nicht angezeigt), wird es in der Regel versuchen, das Objekt einer bekannten Objektklasse zuzuordnen. Um dies zu verhindern, sollten Objekte, die in der Anwendung vorkommen können, auch in den Trainingsdaten enthalten sein. Damit lernt das Modell, dass diese Objekte nicht relevant sind.

In einem ersten Schritt werden zur Erstellung eines Datensatzes reale Daten erfasst oder Trainingsdaten mit CAD-Modellen oder KI-Bilderzeugungsmodellen generiert, wenn keine realen Bilder verwendet werden können. In unserem Beispiel haben wir echte Daten verwendet. Dazu können entweder Videos oder Bilder aufgenommen werden. Im folgenden Schritt werden die Daten vorverarbeitet. In unserem Fall arbeitet die Laufzeitanwendung zur Objekterkennung mit Bildern der Größe 640 x 640 Pixel. Wir wollten also das Modell mit Bildern derselben Größe trainieren, aber die Kamera zum Aufnehmen der Bilder unterstützte diese Größe nicht. Aus diesem Grund mussten wir sie auf das richtige Format zuschneiden.

Danach werden die Daten mit einem Label versehen. Die Kennzeichnung von Daten in KI-Anwendungen ist die Zuweisung aussagekräftiger und relevanter Labels oder Tags an verschiedene Elemente oder Beispiele innerhalb eines Datensatzes. Das ist ein wichtiger Schritt beim überwachten maschinellen Lernen, bei dem der Algorithmus von gelabelten Beispielen lernt, um Vorhersagen für neue, unbekannte Daten zu treffen.

Natürlich **gibt es spezielle Tools** (<https://aimodels.org/open-source-ai-tools/data-annotation-labeling-tools-machine-learning-datasets/>), die den Benutzer dabei unterstützen. In unserem Fall müssen die zu erkennenden Objekte, d. h. die verschiedenen Pralinen, in allen Trainingsbildern gelabelt werden. Jede Pralinenorte wird einer eigenen Klasse zugeordnet, damit das Modell sie später auch als eigene Sorte/eigenes Produkt erkennen kann. Diese Informationen werden in einer Textdatei gespeichert. Der Trainingsalgorithmus liest dann das Originalbild und die Textdatei mit den Label-Informationen und kann so die relevanten Objekte auf den Bildern lernen.

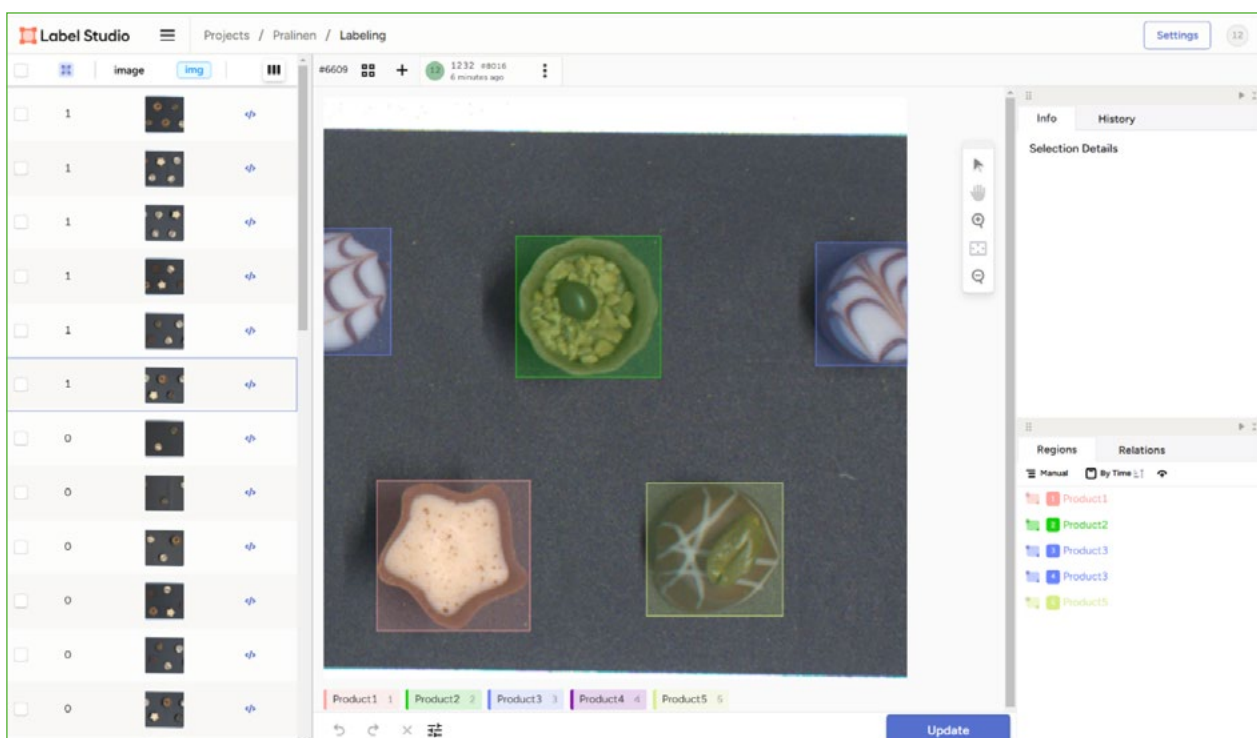


Abbildung 1: Kennzeichnung der Bilder. Im linken Feld können die Bilder ausgewählt werden, für die Labels erstellt werden sollen.

Im mittleren Feld können Labels manuell erstellt werden. Im rechten Feld werden die bereits vorhandenen Labels des aktuellen Bildes dargestellt.

# Bevor wir zum letzten Schritt kommen, können wir mit einem Trick weitere Trainingsdaten erzeugen.

Mittels Dataaugmentation können die bereits gelabelten Trainingsdaten vervielfältigt werden. Dabei müssen die vorhandenen Bilder verändert werden. Solche Änderungen können beispielsweise Bildrotationen, Änderungen der Bildhelligkeit oder Farbtemperatur, Unschärfe, Hinzufügung von Rauschen, Vergrößern oder Verkleinern sein.

Wir verwenden dazu ein eigens erstelltes Skript und können so mit einfachen Bildbearbeitungstechniken die Anzahl der Trainingsbilder um ein Fünffaches erhöhen. So werden aus den ursprünglich 500 Trainingsbildern 2.500. Da diese Operationen auf die bereits gelabelten Daten angewendet werden, können die Labels bei Bedarf auch einfach automatisch angepasst werden. Die Bilder müssen dann nicht erneut gelabelt werden. Um nicht jedes einzelne Bild verändern zu müssen, haben wir Augmentationstechniken nach dem Zufallsprinzip auf die Bilder angewendet, um die gewünschte Anzahl von Bildern zu erhalten.

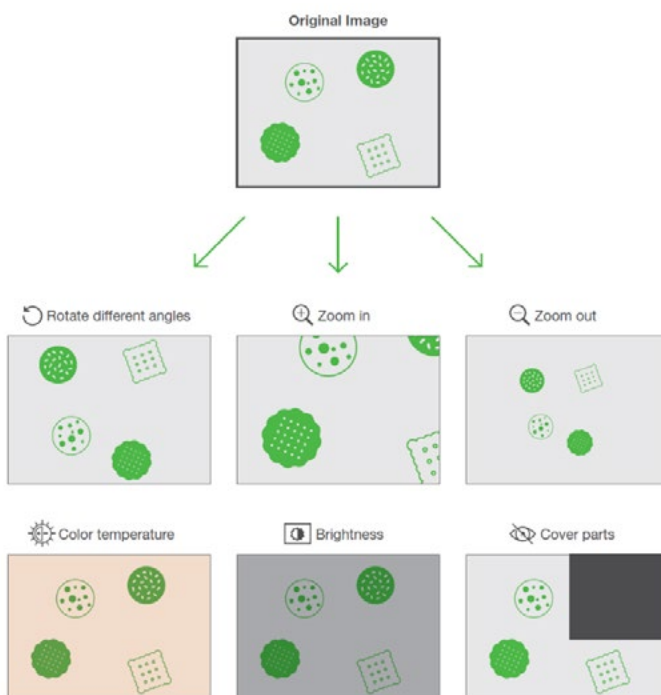


Abbildung 2: Dataaugmentation

Sobald die Daten ausgewählt und vorbereitet wurden, besteht der letzte Schritt darin, den Datensatz in Trainings- und Validierungsdaten aufzuteilen. Auch wenn das Erfassen und Labeln einer großen Datenmenge viel Arbeit macht, lohnt es sich am Ende. Bei strengen Experimenten wird der Datensatz in der Regel in drei Untersätze aufgeteilt: Trainings-, Validierungs- und Testsätze. Der größte Satz, der Trainingssatz, wird für die Hauptaufgabe – das Trainieren des Modells – verwendet.

Der Trainingsfortschritt wird anhand des Validierungssatzes überwacht. Nachdem das Modell trainiert wurde, wird schließlich der dritte Satz verwendet, um die Leistung des Modells unabhängig beurteilen zu können. Das ist wichtig, da der Validierungssatz in der Regel verwendet wird, um zu beurteilen, wie gut das Modell trainiert wurde, wann das Training beendet werden kann und wie Hyperparameter optimiert werden können. Da es jedoch im Prinzip als Optimierungsziel verwendet wird, kann damit keine unabhängige Aussage über die Qualität des Modells getroffen werden.

Eine weitere wichtige Eigenschaft eines guten Trainingsdatensatzes ist die statistische Ausgewogenheit. Es ist wichtig, dass die Klassen gleichmäßig verteilt sind (ähnliche Anzahl von Beispielen aus den einzelnen Klassen), damit das Modell nicht voreingenommen zur Vorhersage der am stärksten vertretenen Klasse tendiert (z. B. hat bei einer Einteilung in zwei Klassen ein Modell, das immer eine Klasse vorhersagt, die 95 % des Datensatzes umfasst, nur 5 % Fehler).

## Haben Sie Fragen?



Vereinbaren Sie einfach einen Termin. Unsere Expertinnen und Experten finden gemeinsam mit Ihnen die optimale Lösung, die perfekt auf Ihr Unternehmen zugeschnitten ist.

### **Ihr Ansprechpartner:**

Stefan Fischereeder  
Produktmanager Industrial AI

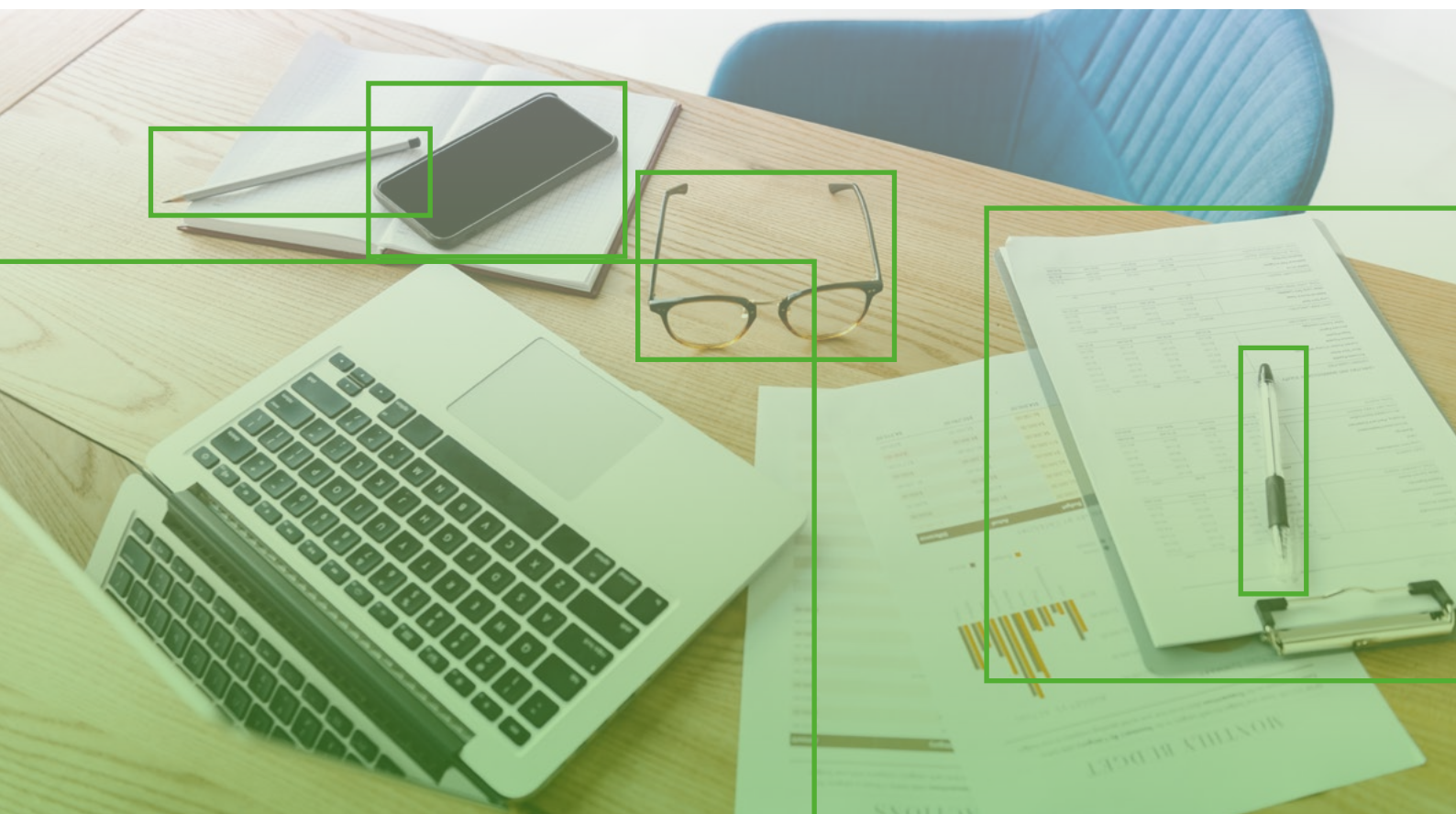
✉ [fsd@keba.com](mailto:fsd@keba.com) ☎ +43 732 7090 22723

## 2. Das Modell trainieren und testen

Sobald der Datensatz fertig ist, kann das Modell trainiert werden. Dazu wird ein Rechner mit einem speziellen Grafikprozessor empfohlen, da dies das Training erheblich beschleunigt. Das Training kann einige Stunden bis zu mehrere Tage dauern. Beim Trainieren eines KI-Modells müssen mehrere Wiederholungen durchlaufen werden, damit gewährleistet ist, dass es ordnungsgemäß aus den Daten lernt und die Parameter schrittweise optimiert. Jeder Abschnitt ermöglicht es dem Modell, den gesamten Datensatz zu berücksichtigen und die internen Parameter durch Optimierungsalgorithmen, z. B. mittels stochastischem Gradientenabstiegsverfahren, anzupassen. Dieser Prozess hilft bei der Generalisierung, sodass das Modell bei neuen, unbekanntem Daten gut funktioniert, indem eine Über- oder Unteranpassung vermieden wird. Komplexe Muster in den Daten erfordern möglicherweise weitere Wiederholungen, damit das Modell sie erfassen und einbeziehen kann.

Nach Abschluss des Trainings wird das Modell anhand realer Daten getestet. Bilder der echten Anwendung werden verwendet, um das trainierte Modell zu testen. Dies ist für den Erhalt gültiger Testergebnisse sehr wichtig. Wie für das Training des Modells gibt es auch für das Testen des Modells eine eigene Anwendung. Das zu testende Modell und die Testbilder werden in diese Anwendung übertragen. Die Anwendung führt dann die Objekterkennung mit den Testbildern durch und speichert die erkannten Bilder. Diese Bilder können danach automatisch von der Anwendung oder manuell vom Entwickler überprüft werden. Die Einschätzung, ob ein Modell gut genug für die Anwendung ist, obliegt in der Regel dem Entwickler. Bei bestimmten Anwendungen muss die Erkennung nicht immer zu 100 % richtig sein, da mögliche Fehler in der Anwendung einfach behoben werden können.

Wenn die Erkennung nicht perfekt sein muss, kann jede Menge Trainingsaufwand gespart werden. In unserem Beispiel dürfen jedoch keine Fehler oder falsch erkannte Objekte auftreten. Daher ist eine sehr sorgfältige Prüfung erforderlich und der Trainingsaufwand ist möglicherweise höher. Sind Sie mit den Ergebnissen zufrieden, d. h. alle Objekte wurden mit der gewünschten Genauigkeit erkannt und es wurden keine Objekte falsch erkannt, können Sie zum nächsten Schritt übergehen und das Modell auf den KI-Chip übertragen. Sind die Testergebnisse noch nicht gut genug, müssen Sie den Trainingsprozess wiederholen und den Datensatz verbessern.



## 3. Das Modell in einer echten Anwendung einsetzen

In einem letzten Schritt wird das von dem Grafikprozessor trainierte Modell auf den KI-Chip übertragen. Das ist notwendig, da das Grafikprozessormodell mit Fließkommawerten arbeitet und der KI-Chip im KEBA Modul nur Ganzzahlen verarbeiten kann. Dies ist keine Besonderheit unseres Moduls, denn fast alle Edge-Beschleuniger verarbeiten nur INT8. Dadurch ist jedoch eine hohe Bilder-pro-Watt-Leistung möglich, damit das Modul ressourcenschonend arbeitet. Die Quantisierung von FP32 zu INT8 führt natürlich zu einem Verlust an Genauigkeit. Daher wird nach der Quantisierung eine Kalibrierung des Modells mit Bildern aus dem Trainingsdatensatz durchgeführt. Dieser Schritt erfolgt automatisch von der vom Chiphersteller bereitgestellten Toolchain.

Nach Abschluss der Quantisierung sollte erneut getestet werden, ob die Erkennung noch mit der gewünschten Genauigkeit erfolgt. Ist dies nicht der Fall, müsste das Modell erneut trainiert werden. Das ist in der Regel nicht notwendig, da der Genauigkeitsverlust dank der fortschrittlichen Algorithmen, die es inzwischen für diesen Prozess gibt, überschaubar ist.

Der letzte Schritt vor dem tatsächlichen Einsatz des Modells ist die Kompilierung für den Chip. Dieser Schritt wird ebenfalls automatisch von der Toolchain durchgeführt. Jetzt kann das Modell gemeinsam mit dem KI-Modul von KEBA eingesetzt werden.

**Möchten Sie wissen, wie das genau funktioniert, empfehlen wir Ihnen ein weiteres Whitepaper, das wir über die Erstellung von Laufzeitanwendungen verfasst haben:**



**Whitepaper:**  
„Erstellen einer Laufzeitanwendung zur Objekterkennung“

## Fazit

Das Trainieren eines KI-Modells zur Objekterkennung ist heutzutage ganz einfach. Der Fortschritt geht über die KI-Funktionen hinaus und umfasst auch benutzerfreundliche Tools und optimierte Prozesse.

In unserem Fall hat ein Auszubildender die gesamte Anwendung erstellt: vom Generieren der Daten und dem Trainieren des Modells bis hin zum Erstellen der SPS-Laufzeitanwendung. Dabei waren unsere KI-Spezialisten und Applikationsingenieure lediglich unterstützend tätig. Das verdeutlicht, dass Fachwissen nicht mehr zwingend erforderlich ist, um KI zu trainieren oder KI-Anwendungen zu entwickeln. Darüber hinaus vereinfachen die tiefe Integration der KI-Algorithmen und -Modelle sowie die Bildverarbeitungsfunktionen der KEBA Steuerung das Erstellen von Anwendungen. Die vollständige KI-Anwendung kann direkt im SPS-Code erstellt werden, was insbesondere für weniger erfahrene Entwickler:innen eine große Erleichterung darstellt. KI bietet eine Vielzahl interessanter Anwendungsfälle für die Bewältigung realer Herausforderungen in industriellen Umgebungen, einschließlich solcher, die bisher als unlösbar galten. KEBA verfügt über die Erfahrung, Kunden bei der Lösung dieser Probleme zu unterstützen und bietet leistungsstarke KI-Inferenz-Hardware, Softwarefunktionen und Unterstützung im technischen Bereich.

## Weitere Informationen



### Produktseite:

AE 550

[www.keba.com/ai-extension-module](http://www.keba.com/ai-extension-module)



### Interview:

AE 550

[www.keba.com/ai-in-the-industry](http://www.keba.com/ai-in-the-industry)



### News & Blog:

Aktuelle Neuigkeiten, Pressemeldungen und interessante Fakten

[www.keba.com/ia-blog-ai](http://www.keba.com/ia-blog-ai)





**KEBA Industrial Automation GmbH**

Reindlstraße 51, 4040 Linz/Austria, Telefon +43 732 7090-0, keba@keba.com

**KEBA Group weltweit**

China / Deutschland / Großbritannien / Indien / Italien / Japan / Niederlande / Österreich  
Rumänien / Schweiz / Serbien / Südkorea / Taiwan / Tschechische Republik / Türkei / USA

[www.keba.com](http://www.keba.com)



Automation by innovation.