

Stressfrei ins neue Schuljahr dank Machine Learning

Im Zuge der Weiterentwicklung der Thalia App entstehen regelmäßig neue Features. Dabei versuchen wir bereits bestehenden Code durch Refactorings für uns, aber auch für den Kunden, zu verbessern. Nach der Pfadfinderregel hinterlässt jeder im Team die Codebasis ein bisschen besser als vorgefunden.

Problemstellung und bisheriger Prozess

Beim Aufräumen des Android Manifest bin ich auf eine Activity (repräsentiert eine Ansicht innerhalb einer App) gestoßen, die sich mit Schulbüchern befasst. Thalia hat jedes Jahr zwischen Juni und September Aktionen für Schulartikel. Dort können Hefte, Kalender oder auch Schulbücher für den Start in das neue Schuljahr gekauft werden. Die Schüler erhalten von ihren Lehrern in der Regel eine Liste mit den benötigten Materialien am Anfang des neuen oder zum Ende des vergangenen Schuljahrs. Eine solche Liste beinhaltet gewöhnlich eine European Article Number (kurz: EAN), einen Titel und einen Preis.

Die Liste kann mithilfe der Kamera über die Thalia App fotografiert und im Anschluss per Mail an Thalia übermittelt werden. Das Foto wird mit weiteren Informationen von der App an einen E-Mail-Client auf dem Gerät übergeben und für den Bestellprozess an schulbuch@thalia.de verschickt. Mit wenigen Klicks und dem Verfassen einer E-Mail waren die benötigten Materialien für das neue Schuljahr bestellt. Der Prozess ist funktional – aber kann dieser für den Kunden noch einfacher gemacht werden?

Idee zur Verbesserung

Gemäß eines unserer Schlüsselverhalten bei Thalia versuchen wir unsere Lösungen spielerisch einfach zu gestalten. Der vorher beschriebene Prozess funktioniert, lässt aber einige Fragen offen: Können die Artikel nur per Rechnung bezahlt werden oder lässt die Bestellung per Mail auch andere Zahlungsmöglichkeiten zu? Wie wird am einfachsten kenntlich gemacht, wenn ein Artikel nicht bestellt werden soll? Wie wird verfahren, wenn das Bild unkenntlich

oder abgeschnitten ist? Grundsätzlich könnten die Bücher auch vom Kunden selbst gesucht und bestellt werden. Dadurch wären die vorherigen Fragen obsolet, aber das Abtippen einer längeren Liste erscheint weniger praktisch als die Aufnahme dieser mit der Kamera. An der Stelle kam die neue Idee ins Spiel: Machine Learning.

Machine Learning oder maschinelles Lernen ist ein Teilgebiet der Informatik und erkennt durch Muster und Training von Algorithmen Muster in Datensätzen. Auf unseren Anwendungsfall übertragen ist der Text auf dem Zettel, der über das Foto aufgenommen wird, unser Datensatz. Der Kern der Idee besteht darin, den Text auf dem Zettel für eine Suche in unserem Sortiment auszulesen. Prädestiniert im Datensatz sind die EANs. Diese sind eindeutig und würden, im Gegensatz zur manuellen Suche (oder Suche über eine Schnittstelle) nach einem Titel, mit Sicherheit den gewünschten Artikel auffinden.

Für die Erkennung von Text auf Bildern stellt Google mit dem Machine Learning Kit (ML Kit) eine Bibliothek für Android und iOS zur Verfügung[\[1\]](#). Das versprochene Ergebnis nach Analyse eines Bildes ist ein Objekt vom Typ `FirebaseVisionText`[\[2\]](#). Dieses ist entweder leer, wenn kein Text erkannt wurde, oder enthält je nach Präsenz des Textes auf dem Bild Textblöcke, die wiederum in Textzeilen aufgeteilt sind, oder einen einzelnen Satz.

Auf den Schulzettel übertragen würden alle EANs, Titel und Preise erkannt und in einem `FirebaseVisionText`-Objekt bereitgestellt werden. Durch eine Filterung ließen sich dann die EANs über einen regulären Ausdruck auslesen. Diese könnten dann via API-Call gesucht und das Ergebnis in einer Liste dargestellt werden. Mit einem Klick wandern die gefundenen Artikel in den Warenkorb und der Nutzer kann selbstständig die benötigten Schulmaterialien bestellen. Die Umsetzung der beschriebenen Theorie erfolgt im nächsten Kapitel.

Implementierung Android

Die Aufnahme des Bildes wird von der Kamera App des Android OS gehandelt. Wir geben dem entsprechenden Intent eine Uniform Resource Identifier (URI) mit, unter der das Bild nach der Aufnahme gespeichert werden soll.

```

val values = ContentValues()
values.put(MediaStore.Images.Media.TITLE, "titel_des_bildes")
outputFileUri = activity?.contentResolver?.insert(MediaStore.Images.Media.EXTERNAL_CONTENT_URI, values)

takePhotoIntent.putExtra(MediaStore.EXTRA_OUTPUT, outputFileUri)
startActivityForResult(takePhotoIntent, REQUEST_SCHOOLBOOK_IMAGE)

```

Abbildung 1: Intent für die Aufnahme durch die Kamera App

Das aufgenommene Bild wird in unserer Analyzer Klasse als Bitmap erstellt und anschließend unter Zuhilfenahme der ML-Kit-Bibliothek zu einem FirebaseVisionImage konvertiert.

```

private fun prepareImage(imageURI: Uri, rotation: Float): FirebaseVisionImage {
    val matrix = Matrix()
    matrix.postRotate(rotation)
    val original : Bitmap! = Media.getBitmap(context.contentResolver, imageURI)
    val bitmap : Bitmap! = Bitmap.createBitmap(original, 0, 0, original.width, original.height, matrix, filter: true)
    return FirebaseVisionImage.fromBitmap(bitmap)
}

```

Abbildung 2: Konvertierung in FirebaseVisionImage

Dabei wird auch die ursprüngliche Rotation zum Bild hinzugefügt. Die Konvertierung ist erforderlich, damit die Erkennung des Textes anschließend stattfinden kann. Das vorbereitete Bild wird daraufhin mittels eines FirebaseVisionTextRecognizer analysiert und gibt uns das im Theorieteil erwähnte FirebaseVisionText-Objekt zurück.

```

private fun decodeText(firebaseVisionImage: FirebaseVisionImage, result: ScanCallback) {
    val detector : FirebaseVisionTextRecognizer = FirebaseVision.getInstance().onDeviceTextRecognizer
    detector.processImage(firebaseVisionImage)
    .addOnSuccessListener { firebaseVisionText : FirebaseVisionText! ->
        // replace all hyphens and whitespaces
        val scannedText : String = firebaseVisionText.text.replace( oldValue: "-", newValue: "")
        val results : List<String> = regex.findAll(scannedText).toMutableList().map { it.value }
        if (results.isNullOrEmpty()) {
            result.onScanNextRotation()
        } else {
            result.onSuccess(SchoolbookResultObject(cleanScanningResults(results)))
        }
    }
}

```

Abbildung 3: Analyse des aufgenommenen Bildes

Aus dem gescannten Text löschen wir die Bindestriche. Teilweise werden diese auf den Schullisten mit dargestellt, sind für die Suche aber unwichtig. Der bereinigte Text wird durch den regulären Ausdruck überprüft. Dabei werden nur Strings herausgefiltert, die Zahlen zwischen null und neun enthalten und eine Länge von zehn bis 13 besitzen. Die Längen ergeben sich aus der Länge der EAN mit 13 Stellen und der Internationalen Standardbuchnummer (kurz ISBN), die

zehn (ISBN-10) oder 13 (ISBN-13) Stellen besitzt. Diese kann analog zur EAN für unsere Suche genutzt werden und wird teilweise auch auf den Schullisten abgedruckt. Die für die Suche unwichtigen Bindestriche stellen den Unterschied zwischen der EAN und einer ISBN-13 dar.

Anpassungen nach Rollout Android

Nachdem das erste Minimum Viable Product fertiggestellt und bereits in einer neuen App Version veröffentlicht wurde, haben wir noch weitere Verbesserungen vorgenommen.

Eine Maßnahme bestand darin, die ausgelesenen EANs (oder ISBNs) auf ihre Länge hin zu untersuchen.

```
when (current.length) {
    13 -> cleanedList.add(current)
    10 -> cleanedList.add(current)
    11 -> if (current.first() == '8') {
        cleanedList.add(current.removePrefix( prefix: "8"))
    } else if (current.first() == '9') {
        cleanedList.add(current.removePrefix( prefix: "9"))
    }
    12 -> if (current.substring(0, 2) == "78") {
        cleanedList.add(current.removePrefix( prefix: "78"))
    } else if (current.substring(0, 2) == "79") {
        cleanedList.add(current.removePrefix( prefix: "79"))
    }
}
```

Abbildung 4: Bereinigung der gefundenen Ergebnisse

Bei den Längen zehn und 13 können wir uns sicher sein, dass die Nummer korrekt vom Bild ausgelesen wurde. Tauchen im Ergebnis Strings mit den Längen elf oder zwölf auf, wurden entweder ein oder zwei Zeichen zu wenig erkannt (für eine ISBN-13 oder eine EAN) oder es wurden ein oder zwei Zeichen zu viel erkannt (für eine ISBN-10). Die ersten drei Stellen bei einer ISBN-13 (oder EAN) im Kontext der Schulbuchzettel bestehen aus den Zahlenfolgen „978“ (oder „979“). Dabei handelt es sich um Präfixe für das Medium Buch. Da auf den

Schulbuchzetteln in der Regel nur Bücher zu finden sind und wir nicht sicher sein können, ob zu viel oder zu wenig erkannt wurde, löschen wir das Präfix im untersuchten Text. Allerdings auch nur, wenn das Präfix darauf hindeutet, dass eine oder zwei Stellen der ISBN nicht erkannt wurden. Innerhalb der drei möglichen Ziffernfolgen (ISBN-13, EAN und ISBN-10) handelt es sich bei der letzten Stelle um eine Prüfziffer. Durch ein Streichen des vorangestellten Präfixes ist die Prüfziffer nicht korrekt. Die mit dieser fehlerhaften Nummer angesprochene Schnittstelle übernimmt die Fehlerkorrektur, sodass wir keine Prüfziffer neu berechnen müssen. Bei Nummern mit weniger als zehn oder mehr als 13 Stellen hat die Analyse zu sehr gestreut und es wird keine Bereinigung vorgenommen.

Eine weitere Verbesserung haben wir bezüglich der Rotation entwickelt. Das Problem bestand darin, dass die Analyse des Textes auf dem Bild weniger Erfolg hat, wenn dieses z. B. um 90 Grad gedreht war. Das heißt, das Bild wurde z. B. im Landscape-Modus aufgenommen und die Analyse hat im Portrait-Modus ohne Rotation in diesem stattgefunden.

```
if (rotation >= 360) {  
    emitter.onError(Exception("No EAN/ISBN numbers found."))  
} else {  
    rotation += 90f  
    decodeText(prepareImage(imageURI, rotation), result: this)  
}
```

Abbildung 5: Rotation des Bildes für weitere Analyse

Die Lösung für das Problem besteht darin, das Bild solange zu rotieren, bis Ergebnisse gefunden werden. Wenn die Analyse ohne Ergebnis bleibt findet eine erneute Analyse mit einem um 90 Grad rotierten Bild statt. Dieser Vorgang wird so oft wiederholt, bis die Rotation das Bild einmal komplett um 360 Grad gedreht hat oder bis ein Ergebnis vorliegt.

Scannen des Schulzettels in der App

Nach der Theorie und der Beschreibung des Codes soll in Form von Screenshots das Feature in der App gezeigt werden. Als erstes erfolgt die Aufnahme des Bildes.

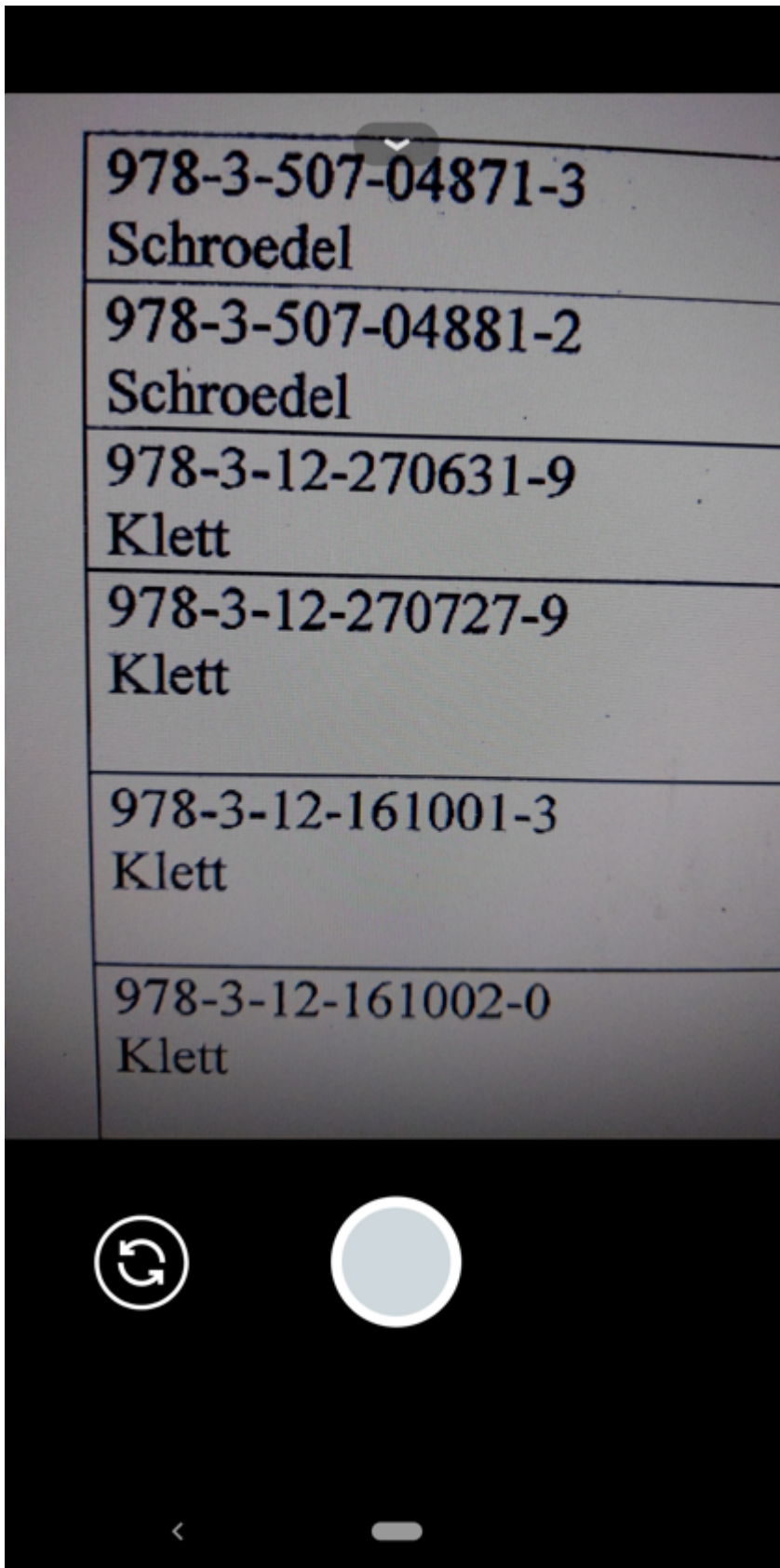


Abbildung 6: Aufnahme des Bildes mit der Kamera App

Anschließend wird das Bild analysiert und die Ergebnisse in einer Liste dargestellt.

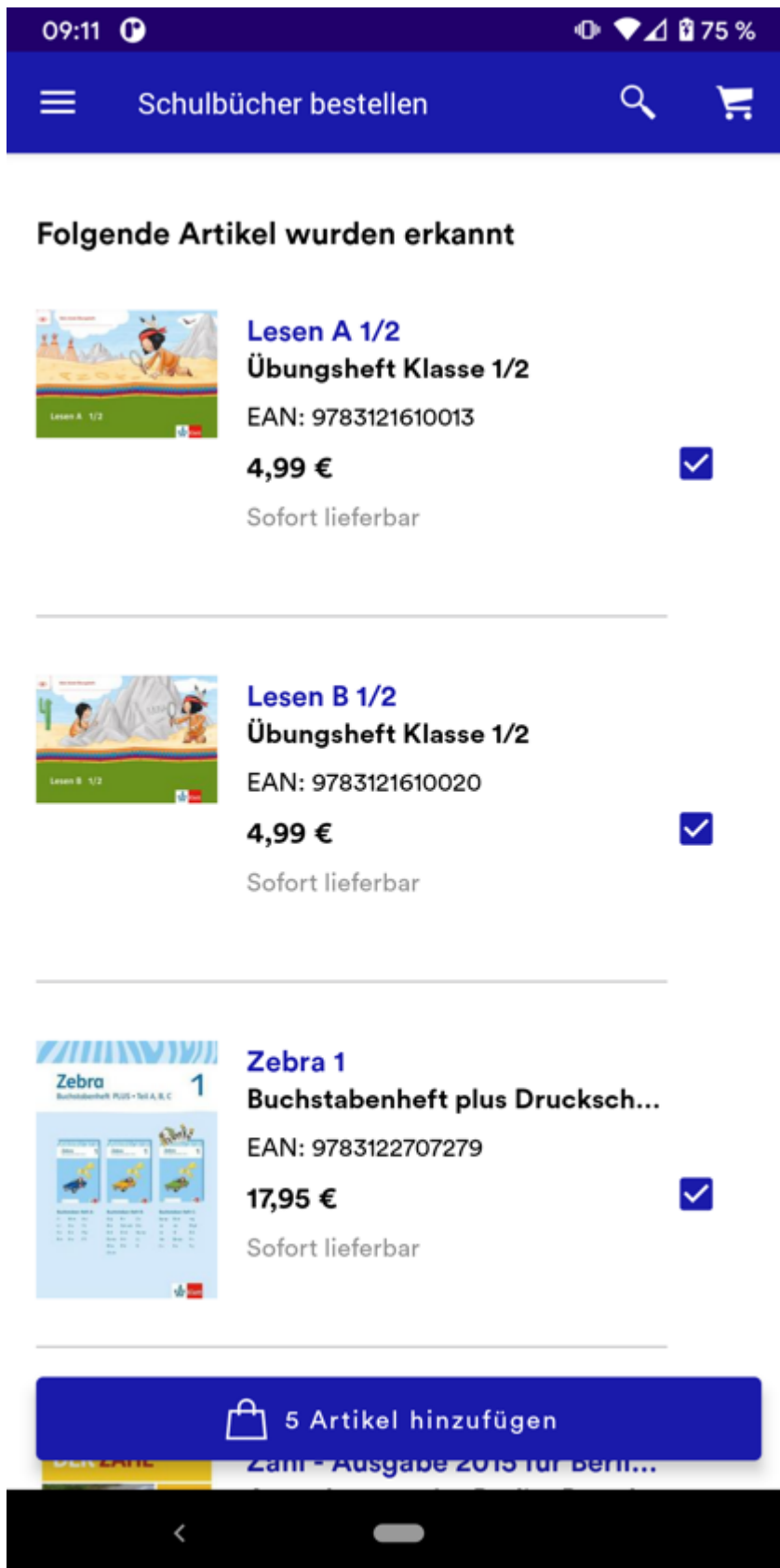
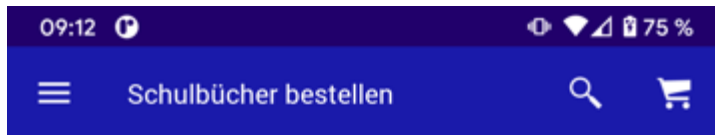


Abbildung 7: Ergebnisliste nach der Analyse des aufgenommenen Bildes
Der Kunde hat die Möglichkeit, die benötigten Materialien in den Warenkorb zu legen.



Alternative Bestelloptionen

Nutzen Sie eine unserer folgenden Alternativen zur bequemen Bestellung ihrer Schulbücher.



E-Mail

Senden Sie eine E-Mail mit einem Foto des Bestellzettels und Ihrer Lieferadresse an:
schulbuch@thalia.de



WhatsApp

Schicken Sie eine Nachricht oder Bild mit ihrer Bestellung, gewünschter Lieferadresse und E-Mail an folgende Nummer:
[0151 64 67 18 61](tel:015164671861).



Schulbuch-Hotline

Rufen Sie Ihren persönlichen Schulbuch-Berater an:
[0251 5 30 93 88](tel:02515309388).

Montag bis Freitag von 10:00 bis 18:00 Uhr



Thalia in ihrer Nähe

Mit persönlicher Beratung von Ihrem Buchhändler vor Ort.

[> Filiale finden](#)



Abbildung 8: Alternativoptionen bei fehlerhafter Analyse

Falls bei einem Scan etwas schief läuft und keine Ergebnisse gefunden werden, sind Alternativen verfügbar. Eine davon ist der eingangs beschriebene Prozess mit der Versendung des Bildes per E-Mail.

Fazit und Ausblick

Der Schulbuchscanner hat den Bestellprozess für Schulmaterialien anhand einer Liste deutlich erleichtert. Für den Kunden ist kein mühsames Eingeben der EANs

oder das Versenden eines Bildes erforderlich. Mit dem Scanner und der Analyse eines Fotos können alle benötigten Materialien unter Nutzung der gewünschten Bezahlmethode direkt an die Wunschadresse bestellt werden. Auch die Mitarbeiter bei Thalia profitieren. Dank des Scanners kommen deutlich weniger Anfragen bei schulbuch@thalia.de an, wodurch eine geringere Anzahl an Bestellungen für die Kunden manuell vorgenommen werden muss.

Künftig könnte die generelle Scanning-Funktionalität auch für andere Use-Cases verwendet werden. Die Erkennung von Text auf Bildern ist, wie der Schulbuchscanner gezeigt hat, möglich. Allerdings wird der Anwendungsfall hier dadurch erleichtert, dass ein Großteil des Textes durch einen regulären Ausdruck herausfilterbar ist.

[1] <https://developers.google.com/ml-kit/vision/text-recognition>

[2]

<https://firebase.google.com/docs/reference/android/com/google/firebase/ml/vision/text/FirebaseVisionText>