**T.C.**

**TÜRKISCH-DEUTSCHE UNIVERSITÄT**

**FAKULTÄT FÜR NATURWISSENSCHAFTEN**

****

**MOLEKULARE BIOTECHNOLOGIE**

**Ein Kanalbasierter Optimizierungsansatz zur Erkennung von Tomatenkrankheiten**

**BACHELORARBEIT  
  
Yusuf YILDIZ  
200102043**

**ISTANBUL, 2025**

**T.C.**

**TÜRKISCH-DEUTSCHE UNIVERSITÄT**

**FAKULTÄT FÜR NATURWISSENSCHAFTEN**

****

**MOLEKULARE BIOTECHNOLOGIE**

**Ein Kanalbasierter Optimizierungsansatz zur Erkennung von Tomatenkrankheiten**

**BACHELORARBEIT  
  
Yusuf YILDIZ  
200102043**

**ISTANBUL, 2025**

**T.C.**

**TÜRKISCH-DEUTSCHE UNIVERSITÄT**

**FAKULTÄT FÜR NATURWISSENSCHAFTEN**

**MOLEKULARE BIOTECHNOLOGIE**

**EIN KANALBASIERTER OPTIMIZIERUNGSANSATZ ZUR ERKENNUNG VON TOMATENKRANKHEITEN**

Yusuf YILDIZ

**BACHELORARBEITSKOMITEE:**

Assist. Prof. Dr. Dilek GÖKSEL DURU …………………

(Betreuerin)

Prof. Dr. Orkide COŞKUNER WEBER …………………

Prof. Dr. Hasan Birol ÇOTUK …………………

**Abgabedatum:** 30/05/2022

EIGENSTÄNDIGKEITSERKLÄRUNG

Ich verspreche, dass ich von mir angeforderte und bei mir eingerichtete Abschlussarbeit ohne fremde Hilfe abfassen werde. Ich versichere, die selbstständig gearbeitet zu haben.

In diesem Thema habe ich die Arbeit selbst oder Abschnitte davon nicht aus vorher veröffentlichten Studienarbeiten entnommen. Alle Zitationen und Quellen, die für meine Arbeit genutzt habe, und, von denen ich profitiert habe, sind als Hilfsmittel in der Stelle “Literaturverzeichnis“ angegeben.

Meine Arbeit wurde von mir noch in keiner Institution oder Organisation mit den angegebenen Inhalten präsentiert.

Unterschrift:

Yusuf Yıldız

23/06/2025

kURZFASSUNG

**Ein Kanalbasierter Optimizierungsansatz zur Erkennung von Tomatenkrankheiten**

Aufgrund des zunehmenden Klimawandels sind landwirtschaftliche Flächen stark gefährdet. Mit dem Fortschreiten des Klimawandels nehmen auch die Probleme zu, mit denen Pflanzen konfrontiert sind. Eine frühzeitige Erkennung dieser Probleme sowie der Aufbau entsprechender Systeme ist daher von großer Bedeutung. In diesem Zusammenhang kommt der Tomatenpflanze ein besonderer Wert zu. Tiefe Lernmodelle verfügen über eine hohe Fähigkeit, Probleme oder Krankheiten anhand von Pflanzenbildern (insbesondere von Blättern) erfolgreich zu erkennen. Aus diesem Grund konzentriert sich diese Studie auf ResNet-Modelle, eine Art von Deep-Learning-Architektur. Zur Umsetzung wurden 9 Bilder von kranken Tomatenblättern und 1 Bild eines gesunden Blattes aus dem PlantVillage Datensatz verwendet. Zwar wurden in der Vergangenheit ähnliche Studien durchgeführt, jedoch zielt diese Forschung darauf ab, ein erfolgreiches, schnelleres und leichteres Modell zu entwickeln. Drei verschiedene ResNet-Modelle – ResNet-18, ResNet-34 und ResNet-50 – wurden mit drei unterschiedlichen Trainingsstrategien (10 Epochen, 25 Epochen und Early Stopping) trainiert. Die Kanal-Ausgaben aus den letzten Convolutional-Schichten dieser Modelle wurden anschließend erneut auf dieselben Modelle angewendet, wodurch insgesamt 36 Modelle entstanden (9 aus dem Ersttraining und 27 aus dem Re-Training). Die erfolgreichsten Modelle waren: das mit den Kanälen von ResNet-18 erneut trainierte ResNet-34-Modell (mit Genauigkeiten von 98,9 % bei 10 Epochen und 99,7 % bei 25 Epochen) sowie das mit den Gewichten von ResNet-18 und Early Stopping erneut trainierte ResNet-50-Modell (Genauigkeit: 99,2 %). Aufgrund möglicher Überanpassung (Overfitting) wird jedoch insbesondere für das ResNet-34-Modell, das mit 25 Epochen trainiert wurde, eine weiterführende Forschung und Optimierung dringend empfohlen.

**Schlüsselwörter:** Konvolutionale Neuronale Netzwerke, Kanäleschneidung, Merkmalextrahierung, semantisches Wörterbuch, ResNet, Krankheitserkennung, Klassifikation

ABSTRACT

**An Optimization Approach for Detecting Tomato Diseases Using Channel Pruning**

Due to increasing climate change, agricultural areas are at significant risk. As climate change intensifies, the problems faced by plants also increase. Early diagnosis of these problems and the establishment of systems based on early detection are therefore highly important. In this context, the tomato plant holds particular value. Deep learning models have a strong capacity to successfully detect problems or diseases based on plant images (especially leaf images). For this reason, this study focuses on ResNet models, which are among the deep learning architectures. To set up this framework, 9 diseased tomato leaf images and 1 healthy tomato leaf image from the PlantVillage dataset were used. While similar studies have been conducted in the past, this research aims to develop a successful, faster, and lighter model. Three different ResNet models—ResNet-18, ResNet-34, and ResNet-50—were trained using three different training strategies (10 epochs, 25 epochs, and early stopping), and the channel outputs from their last convolutional layers were then reapplied to the same models to create 36 models in total (9 from the initial training and 27 from the secondary training). Among these, the most successful models were: the ResNet-34 model retrained with channel outputs from ResNet-18 (trained for 10 and 25 epochs, with accuracy rates of 98.9% and 99.7% respectively), and the ResNet-50 model retrained with weights from ResNet-18 using the early stopping method (accuracy: 99.2%). Nevertheless, due to the possibility of overfitting, further research and optimization—especially for the ResNet-34 model trained for 25 epochs—are highly recommended.

**Key Words:** Convolutional Neural Networks, Channel pruning, Feature extraction, semantic dictionary, ResNet, Disease Detection, Classification

ÖZET

**Domates Hastalıklarını Tespit Etmeye Yönelik Kanal Baskılamalı Bir Optimizasyon Yaklaşımı**

Artan iklim değişikliğinden dolayı tarım alanları büyük bir risk altındadır. İklim değişikliği arttıkça bitkilerin karşılaştığı problemler de artmaktadır. Bu sorunların erken teşhisi ve erken teşhise dayalı sistemler kurabilmek bu amaçla fazlasıyla önemlidir. Bu açıdan domates bitkisi önemli bir değere sahiptir. Derin öğrenme modelleri bitkilerden (özellikle yapraklardan) alınan görsellere dayanarak başarılı problem ya da hastalık tespiti yapabilme kapasitesine fazlasıyla sahiptir. Bu sebeple bu araştırmada derin öğrenme modellerinden olan ResNet modellerine odaklanılmıştır. Bunu kurgulayabilmek için PlantVillage veri setinin 9 adet hastalıklı domates yaprak görselleri ve 1 adet sağlıklı domates yaprak görselleri kullanılmıştır. Geçmişte benzer araştırmalar yapılmıştır ancak bu araştırma hem başarılı hem de daha hızlı çalışan ve hafif bir model kurgulamaya odaklanmaktadır. Üç farklı ResNet Modeli olan ResNet-18, 34 ve 50 yine üç farklı eğitim stratejisi ile eğitilmiş (10 Epoch, 25 Epoch ve erken durdurma) ve bu modellerin son konvolüsyonel katmanlarından alınan kanal çıktıları daha sonra aynı modellere tekrar uygulanarak farklı hızlara ve başarılara sahip 36 model oluşturulmuştur (9 ilk eğitim ve 27 ikinci eğitim olmak üzere toplam 36 model). İçlerinden en başarılı modeller 10 ve 25 Epoch için ResNet18 kanalları ile tekrar eğitilmiş ResNet34 Modeli (doğruluk: %98.9 ve %99.7) ve erken durdurma metodu için ResNet18 modelinden alınan ağırlıklarla tekrar eğitilmiş ResNet50 modeli (doğruluk: %99.2) olmuştur. Yine de aşırı öğrenme ihtimallerinden dolayı bu üç modelin, ama özellikle 25 Epochta eğitilmiş ResNet34 modeline ait daha fazla araştırma ve optimizasyon fazlasıyla tavsiye edilmektedir.

**Anahtar Sözcükler:** Konvolsüyonel Nöral Ağlar,Kanal Budama, Özellik Çıkarımı, Semantik Sözlük, ResNet, Hastalık Tespiti, Sınıflandırma

INHALTSVERZEICHNIS

[EIGENSTÄNDIGKEITSERKLÄRUNG iii](#_Toc201497583)

[KURZFASSUNG iv](#_Toc201497584)

[ABSTRACT v](#_Toc201497585)

[ÖZET vi](#_Toc201497586)

[INHALTSVERZEICHNIS vii](#_Toc201497587)

[Abbildungsverzeichnis ix](#_Toc201497588)

[Tabellenverzeichnis x](#_Toc201497589)

[Abkürzungsverzeıchnıs xi](#_Toc201497590)

[Formelnverzeichnis xii](#_Toc201497591)

[Vorwort xiii](#_Toc201497592)

[Danksagung xiv](#_Toc201497593)

[1. Einleitung 15](#_Toc201497594)

[1.1. Ziel 15](#_Toc201497595)

[1.2. Maschinelles und tiefes Lernen 16](#_Toc201497596)

[1.2.1. Algorithmen 16](#_Toc201497597)

[1.2.2. Maschinelles Lernen 16](#_Toc201497598)

[1.2.3. Tiefes Lernen 16](#_Toc201497599)

[1.2.4. Konvolutionale Neuronale Netzwerken 17](#_Toc201497600)

[1.2.4.1. Konvolution 18](#_Toc201497601)

[1.2.4.2. Residuale Netzwerken (ResNet) 18](#_Toc201497602)

[1.3. Semantisches Wörterbuch und Pruning 20](#_Toc201497603)

[1.3.1. Das semantische Wörterbuch 20](#_Toc201497604)

[1.3.2. Pruning (Zurückschneiden) 21](#_Toc201497605)

[1.3.3. Hitzemaps und visualisierung mit Grad-CAM 21](#_Toc201497606)

[1.4. Verwandte Studien 21](#_Toc201497607)

[2. Material und Methoden 23](#_Toc201497608)

[2.1. Daten 23](#_Toc201497609)

[2.2. Modell- und Gerätenauswahl 25](#_Toc201497610)

[2.3. Erstes Training 25](#_Toc201497611)

[2.3.1. Frühstopp Mechanismus 26](#_Toc201497612)

[2.4. Semantisches Wörterbuch 26](#_Toc201497613)

[2.5. Zurückschneiden und zweites Training 27](#_Toc201497614)

[3. Ergebnisse 28](#_Toc201497615)

[3.1. Ergebnisse der ersten Trainings 28](#_Toc201497616)

[3.2. Ergebnisse und Bildtransformation der semantischen Daten 31](#_Toc201497617)

[3.3. Ergebnisse des zweiten Trainings 33](#_Toc201497618)

[4. Diskussion 38](#_Toc201497619)

[4.1. Diskussion 38](#_Toc201497620)

[4.1.1. Dateianzahlen 38](#_Toc201497621)

[4.1.2. Trainingsfehlern 38](#_Toc201497622)

[4.1.3. Vergleich zwischen erwarteten und angenommenen Ergebnissen 39](#_Toc201497623)

[4.1.4. Verglichen mit Literatur 41](#_Toc201497624)

[4.2. Zukünftige Arbeiten 43](#_Toc201497625)

[4.2.1. Modelle verbessern 43](#_Toc201497626)

[4.2.2. Automatisierung mit anderen Systemen und Applizieren im Feld 43](#_Toc201497627)

[5. Literaturverzeichnis 44](#_Toc201497628)

Abbildungsverzeichnis

|  |  |  |
| --- | --- | --- |
| **Abbildung 1.1** | Ein residualer Block. | 19 |
| **Abbildung 2.1** | Dateienbeispiele für jede Klasse aus Datensatz. | 25 |
| **Abbildung 3.1** | Trainings- und Validierungsverlustkurven der Modelle als Trainingsstrategie. | 29 |
| **Abbildung 3.2** | Verwirrungsmatrizen der Modelle als Trainingsstrategie. | 30 |
| **Abbildung 3.3** | Eine Beispielabbildung der Ausgang von einen Kanal aus beiden ResNet18 und ResNet34 von 25 Epoch bei Klasse Bacterial Spot. | 31 |
| **Abbildung 3.4** | Genauigkeitsgraph von Modellen nach Zurückschneidung. | 35 |
| **Abbildung 3.5** | Höchste Genaugkeitswerte nach Zurückschneidung. | 36 |
| **Abbildung 3.6** | Trainings- und Validierungsverlustkurven für 3 erfolgreichste Modelle als ihren Trainingsstrategien. | 37 |

Tabellenverzeichnis

|  |  |  |
| --- | --- | --- |
| **Tabelle 1.1** | Bewertungsmethode Tabelle. | 17 |
| **Tabelle 1.2** | Architekturtabelle zu den Modellen ResNet-18, 34 und 50. | 20 |
| **Tabelle 2.1** | Klassen und Bildzahlen für jede Klass. | 24 |
| **Tabelle 3.1** | Standard-Trainingsergebnis von Modellen. | 28 |
| **Tabelle 3.2** | Die gemeinsamen Kanäle und Zahlen aus drei Modellen von drei Trainingsstrategien. | 32 |
| **Tabelle 3.3** | Zurückgeschnittene (pruned) Trainingsergebnisse von Modellen. | 33,34 |
| **Tabelle 4.1** | Genauigkeitsänderungen, Genauigkeitswerte und Genauigkeitsmittelwerte für 3 ResNet-Modelle mit 3 unterschiedliche Kanäleherkunften. | 40 |
| **Tabelle 4.2** | Tabelle zur Literaturvergleich. | 41,42 |

Abkürzungsverzeıchnıs

|  |  |
| --- | --- |
| **CNN** | Convolutional Neural Networks (Konvolutionale Neuronale Netze) |
| **ResNet** | Residual Networks (Residuale Netze) |
| **RN** | ResNet |
| **RN18** | ResNet-18 |
| **RN34** | ResNet-34 |
| **RN50** | ResNet-50 |
| **TP** | True Positive (richtig positive) |
| **TN** | True Negative (richitg negative) |
| **FP** | False Positive (falsch positive) |
| **FN** | False Negative (falsch negative) |

Formelnverzeichnis

|  |  |  |
| --- | --- | --- |
| **Formel 1** | Präzision | 18 |
| **Formel 2** | Sensitivität | 18 |
| **Formel 3** | Genauigkeit | 18 |
| **Formel 4** | F1-Skor | 18 |
| **Formel 5** | Konvolution | 18 |

Vorwort

In dieser Studie habe ich versucht und geforscht, die Leistung von ResNet-Modellen, die auf maschinellem Lernen basieren, zur Erkennung von Pflanzenkrankheiten zu optimieren, indem ich die Kanäle in der letzten Schicht des Modells mit dem semantischen Wörterbuch-Verfahren blockierte. Um dies zu erreichen, habe ich den PlantVillage-Datensatz verwendet, der ein visueller Datensatz mit Blattbildern von unterschiedlichen Pflanzen ist. Ich habe meine Modelle mit den Bildern von Tomatenkrankheiten in diesem Datensatz trainiert und die semantischen Wörterbücher extrahiert, um sie dann durch Kanalzurückschneidung erneut zu trainieren. Ich glaube, dass ich auf diese Weise interessante Ergebnisse erzielt habe. Während des gesamten Prozesses stand ich vor verschiedenen Schwierigkeiten und die Ergebnisse der Forschung sind tatsächlich ganz anders ausgefallen, als ich es ursprünglich angestrebt hatte, aber ich bin mit allem zufrieden, was ich erreicht habe, und hoffe, dass ich einen bedeutenden Beitrag zur Literatur leisten kann.

Danksagung

Zuerst möchte ich meinem Mentorin Dr. Dilek Göksel DURU danken, der mich mit seinen Anleitungen und dem erfolgreichen Erlernen von maschinellen Lernsystemen unterstützt hat und mir sowohl die Bereiche Signal- als auch Bildverarbeitung nähergebracht hat.

Dann möchte ich mich bedanken bei meinen Freunden, mit denen ich in der gleichen These-Gruppe zusammengearbeitet habe, insbesondere Ece, Kaan und Çağatay, und bei meiner anderen Mentorin von einer anderen Uni, Dozentin Dr. Elif YÜZBAŞIOĞLU, die mich unterstützt hat, obwohl ich das Thesethema nicht in die von mir gewünschte Richtung lenken konnte.

Danach möchte ich meiner Familie und meiner Freundin danken, die mir während des gesamten Prozesses sowohl materiell als auch emotional keine Unterstützung verweigert haben.

Am Ende möchte ich jedoch dem Gründer der Republik Türkei, dem Führer Mustafa Kemal ATATÜRK, für die Möglichkeit, an diesem Tag zu erreichen, meine aufrichtige Danksagung anbieten.

Yusuf YILDIZ

İstanbul, Juni 2025

1. eınleıtung

Der zunehmende Klimawandel und die jeden Tag steigenden Temperaturen zwingen die Menschheit zum Treibhausanbau und zur Verwendung nachhaltiger Produkte. Dürre, unregelmäßige Regen und Temperaturschwankungen, zunehmendes Bevölkerungswachstum und die Notwendigkeit, sich gegen neu auftretende Krankheiten zu schützen, machen, dass die Menschheit auf den Gewächshausanbau umsteigt [1]. Doch die Landwirtschaft auf der ganzen Welt ist immer noch weitgehend klassisch.

Eines der größten Probleme in der Landwirtschaft ist der Rückgang der Ernteerträge aufgrund verschiedener Krankheiten und vielfältiger Probleme, die in einigen Fällen sogar ganz verschwinden. Nach Angaben des türkischen Statistikinstituts beträgt diese Rate im Durchschnitt 3.5 Prozent pro Jahr [2]. Deswegen sind die frühzeitige Bestimmung und Diagnose, und also Bekämpfung von Pflanzenkrankheiten in der Landwirtschaft von kritischer Bedeutung für die Verringerung von Ertragsverlusten und die Steigerung der Produktivität im Produktionsprozess.

* 1. Ziel

In den letzten Jahren wurde bedeutende Forschung gemacht, um die pflanzlichen Krankheiten zu bestimmen und zu identifizieren. Es wurden verschiedene Methoden und ML-Modelle entwickelt, um die Identifizierung von Pflanzenbedingungen in realen landwirtschaftlichen Umgebungen zu unterstützen. In dieser Studie wurden speziell Tomatenpflanzenkrankheiten mithilfe eines CNN-Modells, benannt als ResNet, und des Datensatzes PlantVillage, die Identifizierung und Modellverbesserungen probiert [3]. Die Forschungsfrage dieser Studie ist: ,,Ist es möglich, während die Modellgenauigkeit gehalten wird, die Parameter zu verkleinern und die Berechnungskosten zu verringern?’’. In diesem Zusammenhang werden verschiedene ResNet-Architekturen evaluiert und ein semantischer, wörterbuchbasierter Ansatz implementiert, um sowohl die Klassifizierungsgenauigkeit als auch die Rechenzeit zu optimieren.

* 1. Maschinelles und tiefes Lernen
     1. Algorithmen

Ein Algorithmus ist ein Instrument. Die Folgen bestimmter Regeln und am Ende gibt die Regel das anwendete Produkt. Die Regeln sind meistens mathematische Operationen im Programmierungsfeld. Seit der erste Computer entwickelt wurde, wurden auch die Algorithmen entwickelt und an verschiedene Probleme angepasst [4].

* + 1. Maschinelles Lernen

Maschinelles Lernen ist ein Begriff, der für das Verarbeiten und Lernen mit verschiedenen und vielfältigen Daten mit einem bezeichneten Algorithmus steht. Das maschinelle Lernen entsteht dank der Idee, die Systeme Schritt für Schritt besser die Aufgaben lösen zu lassen. Um es klarer zu sagen: Ein maschinelles Lernmodell beinhaltet viele randomisierte oder blanke Parameter, und auch jeder Parameter kann ein Algorithmus oder eine mathematische Funktion sein. Nach jeder Iteration werden diese Parameter sich verbessern probieren, um die Hauptaufgabe zu lösen [4]. Wie die Algorithmen, maschinellen Lernen Modellen wurden auch entwickelt durch Zeit passt.

* + 1. Tiefes Lernen

Tiefes Lernen ist eine Untergruppe des maschinellen Lernens, die sich mit hierarchischem Merkmalslernen befasst. Es wurde gebaut, über die Idee: „Wie kann man die neuronalen Lernprozesse über den Bereich des maschinellen Lernens applizieren?“ Ein tiefes Lernmodell hat die Fähigkeit, aus Rohdaten die Merkmale zu extrahieren und Schritt für Schritt zu „lernen”, deswegen verbessert es sich. Wegen dieser „Hierarchie“ benennt dieser Algorithmus hierarchisches Merkmalslernen [4].

Eine große Differenz zwischen tiefem Lernen und maschinellem Lernen ist, dass dank ihrer hierarchischen Struktur, tiefes Lernen komplexe Muster und Merkmale besser „lernen“ kann.

Es gibt unterschiedliche tiefes Lernen-Architekturen; Tiefe-Boltzmann-Maschine, Tiefe Glaubennetzwerke, gestapelte Autoencoder und künstliche neuronale Netzwerke (ANN) sind manche, aber die meiste genutzte Tiefen-Lernen-Architektur sind konvolutionelle neuronale Netzwerke (CNN) [4].

* + 1. Konvolutionale Neuronale Netzwerken

Konvolutionale neuronale Netzwerke wurden erstens im Jahr 1998 entwickelt. Dieser Artikel fokussiert ein Modell, das der Forscher entwickelte: LeNet-5. Dieses Modell wurde auf die graustufige handschriftliche Zeichen- und Zeichenerkennung appliziert, der Eingang soll 32 × 32 Pixel groß sein. Erste konvolutionale Schicht (6 @ 28x28) 6 Filter mit Kernelgröße 5x5, S2\_Unterabtastung (6 @ 14x14) Durchschnittliches Pooling C3-Faltung (16 @ 10x10) 16 Filter S4-Unterabtastung (16 @ 5x5). Mehr Pooling C5 vollständig verbunden (120) Abgeflachte + dichte Schicht F6 vollständig verbunden (84) Dichte Schichtausgabe Softmax (10 Klassen) 10 Ziffern: 0 bis 9 [6].

Um den Erfolg eines Modells berechnen zu können, wurden verschiedene Bewertungsmethoden anhand der vier grundlegenden Vorhersagezustände formuliert, die das Modell als Ausgabe liefert. Diese Vorhersagezustände sind der Reihe nach: True Positive (TP) (richtig positiv), True Negative (TN) (richtig negativ), False Positive (FP) (falsch positiv) und False Negative (FN) (falsch negativ). Zur besseren Verständlichkeit sind diese Zustände in der untenstehenden Tabelle 1.1 erläutert [7].

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabelle 1.1:** Bewertungsmethode Tabelle | | | |
|  |  | **Richtig Klass** | |
|  |  | Positive | Negative |
| **Vorhersage** | Positive | TP | FP |
| Negative | FN | TN |

Basierend auf diesen Zuständen gehören die Genauigkeit (Accuracy) und der F1-Score zu den Metriken, die zur Analyse der Modellleistung entwickelt wurden. Einfach ausgedrückt ist die Genauigkeit das Verhältnis der korrekt vorhergesagten Beispiele zur Gesamtzahl der Vorhersagen, während der F1-Score das Gleichgewicht zwischen Präzision (Precision) und Sensitivität (Recall) misst und insbesondere bei unausgewogenen Datensätzen von Bedeutung ist. Die Formeln (1,2,3,4) für diese Metriken sind unten angegeben [7].

* + - 1. Konvolution

Konvolution ist ein mathematischer Prozess. Konvolution ist einer der sehr populären Prozesse, die in den Feldern der Mathematik, Signalverarbeitung und des maschinellen Lernens genutzt werden. Am einfachsten ist die Konvolution der Prozesse, die zwei Funktionen, die übereinandergesetzt werden, und eine neue dritte Funktion liefert (Formel 5).

In dieser Formel sind f und g die Funktionen, x ist der Eingangs­wert für beide Funktionen, y ist die Versetzungsmenge, und die Grenzen des Integrals sind in der Regel der gesamte Bereich, in dem die Funktionen definiert sind.

* + - 1. Residuale Netzwerken (ResNet)

Der Historie der Resnet wurde über die Probleme ,,Mit zunehmender Schichtdicke steigen die Schwierigkeiten bei der Ausbildung und das Problem der ,,Degradation’’ zwischen den ,,Schichten’’ angefangen. Ende 2015 schlug ein Forschungsteam von Microsoft Research Asia unter Kaiming He, Xiangyu Zhang, Shaoqinq Ren und Jian Sun die Idee des „Residual Lernens“ vor, um dieses Problem zu überwinden [8].

Jeder residuale Block fügt die Eingangsaktivierung nach einer mehrschichtigen Transformation direkt zur Ausgabe hinzu (Skip Connection), um den kontinuierlichen Informationsfluss zu gewährleisten und das Verschwinden der Gradienten mit zunehmender Tiefe der Schichten zu verhindern. Ein allgemeiner ResNet erfasst zunächst mit einer Vielzahl von Konvolutions- und Poolingschichten Merkmale niedriger Komplexität, vertieft dann die hierarchische Darstellungskraft durch aufeinanderfolgende Stufen, die aus unterschiedlichen Residualblöcken (Abbildung 1.1) bestehen, und führt schließlich eine Klassifizierung mit einer Pooling- und einer vollständig verbundenen Schicht durch [5, 7, 8].

|  |
| --- |
| metin, diyagram, yazı tipi, ekran görüntüsü içeren bir resim  Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir. |
| **Abbildung 1.1**: Ein residualer Block [8]. |

Das Arbeitsprinzip basiert auf der Addition des in jedem Block gelernten „residualen“ Funktionswerts mit dem direkt kopierten Eingang. Wenn die Schichten eines Blocks keine komplexere Transformation aus dem Eingang lernen, garantiert das Modell, dass auch tiefere Netze zumindest oberflächlich eine Abbildung durchführen, indem es die Identitätskonnekte direkt bevorzugt. Auf diese Weise verhalten sich tiefere Netze stabiler in Bezug auf Gradiententransport und Fehlerrückpropagation, und effektives Lernen findet auch in mehrschichtigen Architekturen statt [5,7].

Die ResNet-18-, ResNet-34- und ResNet-50-Modelle bestehen in Bezug auf die Schichttiefe aus 18, 34 bzw. 50 Schichten. In ResNet-18 bestehen 8 residuelle Blocken, für ResNet34 bestehen 16 residuelle Blocken und ResNet50 bestehen auch 16 residuelle Blocken. Aber ein Unterschied bildet sich bei ResNet-50-Modell als andere Modelle: Es bietet dreistufige collapse-expand (bottleneck) - Blöcke anstelle von Basisblöcken (1x1, 3x3, 1x1 Faltung) bevorzugt. Die Stufenzahlen für alle drei Modelle sind gleich: 3, 4, 6 und 3 Blöcke. Für das ResNet-50-Modell steigt die Tiefe auf 50 Schichten, weil jeder Block drei Faltungsschichten enthält [8]. In der Tabelle 1.2 stehen die Darstellungen zur Architektur der ResNet-18, 34 und 50 Modelle.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabelle 1.2:** Architekturtabelle zu den Modellen ResNet-18, 34 und 50. | | | | | | |
| **Name der Schicht** | **ResNet-18** | | **ResNet-34** | | **ResNet-50** | |
| conv1 | 7×7, 64, stride 2 | | | | | |
| maxpool | 3×3 max pool, stride 2 | | | | | |
| conv2\_x | [3×3, 64] [3×3, 64] | × 2 | [3×3, 64] [3×3, 64] | × 3 | [1×1, 64] [3×3, 64] [1×1, 256] | × 3 |
| conv3\_x | [3×3, 128] [3×3, 128] | × 2 | [3×3, 128] [3×3, 128] | × 4 | [1×1, 128] [3×3, 128] [1×1, 512] | × 4 |
| conv4\_x | [3×3, 256] [3×3, 256] | × 2 | [3×3, 256] [3×3, 256] | × 6 | [1×1, 256] [3×3, 256] [1×1, 1024] | × 6 |
| conv5\_x | [3×3, 512] [3×3, 512] | × 2 | [3×3, 512] [3×3, 512] | × 3 | [1×1, 512] [3×3, 512] [1×1, 2048] | × 3 |
| avgpool,fc | average pool, 512-d fc | | | | average pool, 2048-d fc | |

In der Tabelle 1.2 stellt die Architekturen für drei unterschiedliche ResNet-Modelle (ResNet-18, 34 und 50) dar, um unterschiedliche Blöcke mit unterschiedliche Schichten besser zu verstehen.

* 1. Semantisches Wörterbuch und Pruning
     1. Das semantische Wörterbuch

Das semantische Wörterbuch ist eine der populärsten Methoden, die in vielen verschiedenen Bereichen eingesetzt wird. Aber in dem Feld des tiefen Lernens wurde die semantische Wörterbuchmethode verwendet, um die Kanalausgänge auf der GAP (Global Avarage Pooling / Globales Durchschnitsspooling) Schichten in CNN-Modellen semantisch zu interpretieren, und zeigt auf, welche Kanäle für die Modellvorhersage wichtig (und wie wichtig) sind. Diese Methode erzeugt kanalbasierte Beitragswerte, indem die Ausgaben der GAP-Schicht mit den Gewichten der FC-Schicht (Fully Connected / vollständig verbunden) multipliziert werden, und bestimmt die semantische Bedeutung der Kanäle. Damit können Sie feststellen, welche Art von Merkmalen der Daten jeder Kanal erfasst, und Sie können wichtige Kanäle auch visuell analysieren [10].

* + 1. Pruning (Zurückschneiden)

Pruning (Zurückschneiden) ist der Vorgang, bei dem unnötige oder wenig beitragende Gewichte oder Neuronen (z.B. Filter, Kanäle oder Verbindungen) in tiefen Lernmodellen entfernt werden. Dadurch wird die Größe des Modells reduziert, die Berechnungskosten sinken, deswegen läuft es schneller [11].

* + 1. Hitzemaps und visualisierung mit Grad-CAM

Heatmap-Grafiken sind eine allgemeine Bezeichnung für Grafiken, bei denen Bereiche mit größerer Bedeutung in den Daten in Rot und Bereiche mit geringerer Bedeutung in Blau dargestellt werden. Grad-CAM ist eine Methode zur Visualisierung aus neuronalen Netzen. Die Aktivierungen der letzten Konvolutionsschichten des Modells werden gewichtet mit Gradienten einer bestimmten Klasse und visualisiert [12]. Mit Grad-CAM werden die Daten von ML/DL-Modellen mithilfe von Heatmap-Grafiken visualisiert und die Bereiche, die für die Entscheidungsfindung von größerer Bedeutung sind, werden visualisiert.

Durch die Benutzung dieser drei Methoden können die Genauigkeit und die Effizienz des Modells steigen. Durch die Berechnung von Wichtigkeitskoeffizienten pro Kanal wird eine objektive Grundlage für das Zurückschneiden von Kanälen mit geringer Wirkung geschaffen, was die Leistung des Modells verbessern kann, also am Ende Kanäle visualisieren.

* 1. Verwandte Studien

In den letzten Jahren wurden viele Studien mit maschinellen Lernmethoden und bildgebenden Verfahren zur Erkennung von Pflanzenkrankheiten mit verschiedenen CNN-Modellen durchgeführt.

In der Studie der Forscher von Sladojevic und Freunde im Jahr 2016 wurde ein auf dem CNN basierendes Modell CaffeNet entwickelt, um 13 verschiedene Pflanzenkrankheiten und gesunde Blätter anhand von Blattbildern zu klassifizieren. Die Daten wurden von Internetquellen gesammelt und mit unterschiedlichen Augmentationsmethoden verbessert. Über der CaffeNet-Architektur wurde das Fine-Tuning angewendet und eine Genauigkeit von 96.3% erreicht [13].

In einer anderen Studie wurden vortrainierte CNN-Architekturen wie VGG-16, VGG-19, ResNet und Inception V3 verglichen, um Tomatenpflanzenkrankheiten an ihren Blättern zu erkennen. Die Daten waren nicht nur ein ausgewogener Datensatz. Die im Labor gesammelten Daten wurden nicht nur im Labor, sondern auch auf dem Feld mit einem Mobiltelefon erreicht, und diese Daten wurden augmentiert und Parameterpassungen wurden durchgeführt. Danach wurden die Modelle dank dieses bereitgestellten Datensatzes trainiert und bewertet. Als Ergebnis zeigte das Inception V3 Modell die beste Leistung in allen Metriken, während Felddaten mit einer durchschnittlich zehn bis fünfzehn Prozent niedrigeren Erfolgsrate als Labordaten für reale Anwendungen anspruchsvoller waren [14].

In einer anderen Studie wurde ein auf einem kostengünstigen und energieeffizienten System arbeitender CNN-basierter Bilddetektor entwickelt, um die pflanzlichen Krankheiten aus ihren Blättern rechtzeitig zu erkennen und zu klassifizieren. Als Methode wurde das CNN-Modell mit den ESCA- und PlantVillage-Datensätzen trainiert und dann mit Pruning komprimiert und in das OpenMV Cam H7 Plus-Gerät integriert. Am Ende zeigt die Studie, dass das entwickelte System effektiv mit eingebetteten Geräten mit begrenzten Ressourcen arbeitet und dabei eine hohe Genauigkeit und schnelle Reaktionszeiten aufweist. Die Genauigkeiten für den ESCA-Datensatz sind 98.10% und für PlantVillage sind es 95.24% [14, 15] .

In einer anderen Studie wurde eine neue kognitive Sehmethode zur automatischen Erkennung von Pflanzenkrankheiten untersucht. Die vorgeschlagene Methode beinhaltet die Verwendung eines fortschrittlichen künstlichen neuronalen Netzwerks (FKNN) zur Bildsegmentierung und anschließend eines auf CNN basierenden Modells zur Klassifizierung der segmentierten Bilder. Diese Methode erzielt eine beeindruckende Leistung mit komplexen und unterschiedlichen Hintergrundbedingungen, eine durchschnittliche Genauigkeit von 93.75% und eine Rückrufquote von 100% [17].

1. Material und methoden
   1. Daten

Die Daten wurden aus dem PlantVillage Datensatz gesammelt [3]. Die erste Version des Datensatzes wurde von Hughes und Salathe 2015 bereitgestellt und dann im Jahr 2019 bei Geetharamani G. und Arun Pandian J. ausgebreitet, es ist also erreichbar im Mendeley und die Daten wurden bei Mendeley Adresse gesammelt [17, 18]. Dieser Datensatz ist einer der größten Datensätze und fokussiert sich auf Pflanzenblätter. In dem Datensatz gibt es 39 Klassen mit 61486 Bildern von gesunden und ungesunden Blättern. Es gibt zwei verschiedene Typen des Datensatzes, einen augmentierten und einen nicht augmentierten. Für diese Studie wurden der augmentierte Typ und nur die Tomatendaten benutzt. Für Tomatenbilder gibt es 10 Klassen mit 18835 Dateien, die Klassen sind also im Tabelle 2.1 zu sehen.

Die Unterschiede zwischen augmentierten und nicht augmentierten Datensätzen wirken sich signifikant auf die Genauigkeit von Deep-Learning-Modellen aus, wie G. & J. (2019) zeigen. Deswegen wurde bei dieser Studie die augmentierte Version benutzt. Um die Augmentation zu machen, wurden sechs verschiedene Augmentationstechniken verwendet. Sie sind Bildumkehr (image-flipping), Gammakorrektur (gamma correction), Rauschinjektion (noise injection), PCA-Farbverstärkung(PCA-color augmentation), Rotation und Skalierung. Am Ende liegen alle Dateien in einer Auflösung von 256×256 Pixeln vor.

Nach der Sammlung die Dateien wurden zwei Untergruppen im Verhältnis 80:10:10 aufgeteilt, um Train-, Test- und Validierungsordner zu unterteilen. Wegen der augmentierten Dateien wurden keine Augmentationsmethoden appliziert, nur die Dateien wurden zu Matrizen transformiert und normalisiert mit ImageNet Mittelwert und standardischer Abweichung für RGB-Werte [20].

|  |  |
| --- | --- |
| **Tabelle 2.1**: Klassen und Bildzahlen für jede Klass | |
| **Klassen** | **Dateianzahl** |
| Spider mites Two-spotted spider mite | 1676 |
| Tomato mosaic virus | 1000 |
| Target Spot | 1404 |
| healthy | 1591 |
| Late blight | 1909 |
| Tomato Yellow Leaf Curl Virus | 5357 |
| Leaf Mold | 1000 |
| Early blight | 1000 |
| Septoria leaf spot | 1771 |
| Bacterial spot | 2127 |
| **total** | **18835** |

In diesem Datensatz ist die Gruppe mit den meisten Daten Tomato Yellow Leaf Curl Virus (5357), während es drei Gruppen mit den geringsten (1000) Daten gibt (Tomato mosaic virus, Leaf Mold, Early blight).

|  |
| --- |
| **yaprak, bitki, doğa içeren bir resim  Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.** |
| **Abbildung 2.1:** Dateienbeispiele für jede Klasse aus Datensatz. |

In der Abbildung 2.1 gibt es Beispiele der Daten aus jeder Klasse. Die Klassennamen für *a)* Bacterial spot, *b)* Early Blight, *c)* Healthy, *d)* Late Blight, *e)* Leaf Mold, *f)* Septoria Leaf Spot, *g)* Spider Mites (Two-spotted spider mite), *h)* Target spot, *i)* Tomato Mosaic Virus, *j)* Tomato Yellow Leaf Curl Virus.

* 1. Modell- und Gerätenauswahl

Zur Wahl des richtigen Modells wurden erstens unterschiedlich tiefe und maschinelle Lernmodelle probiert, viele Artikel gelesen und am Ende die ResNet-18-, ResNet-34- und ResNet-50-Modelle ausgewählt.

Zunächst wurde versucht, das eigene Gerät mit einem Intel Core i5 10210U (1,6 GHz) für die Verarbeitung zu verwenden, jedoch erwies es sich aufgrund seiner begrenzten Leistung als ineffizient. Aus diesem Grund wurden anschließend die Nvidia-L4-GPUs über Google Colab eingesetzt, um eine leistungsfähigere Umgebung zu gewährleisten.

* 1. Erstes Training

Um erste Trainings erfolgreich zu laufen, wurde bei jedem Bild die Eingabegröße des jeweiligen Modells von 224x224 skaliert, als Batch-Size wurde 32 genutzt. Um die Daten in der GPU mit Pytorch zu analysieren, wurden die Daten in Tensorformat umgewandelt. Als Verlustfunktion wurde CrossEntropyLoss gewählt. Dann wurde die timm-Paketbibliothek (Package Library, die in Python enthalten ist) zur Installierung von ResNet-Modellen verwendet. Nicht nur ist diese Funktion die meistgenutzte und populärste, sondern auch Goodfellow und Freunde (2016) zeigen, dass sie erfolgreich bei Klassifikationsproblemen ist. Die Daten wurden mit 3 Modellen und drei verschiedenen Epoch-Arten (10 Epoch, 25 Epoch und Frühstopp-Mechanismus) trainiert. Die Scores der Modelle (wie Genauigkeit, Präzision, Rückruf und F1) und Grafiken (Zug-, Validierungsverlust und Verwirrungsmatrix) wurden einzeln aufgezeichnet. Am Ende des Modelltrainings wurden alle trainierten Modelle im „.pth“-Format aufgezeichnet.

* + 1. Frühstopp Mechanismus

In dieser Studie verwendet Frühstopp Mechanismus als dritte Trainingsstrategie, die darauf abzielt, das Training zu stoppen, wenn der Validierungsverlust des Modells über einen längeren Zeitraum nicht verbessert wird. Am Ende jeder Epoche wird der Validierungsverlust berechnet und mit dem vorherigen Minimalwert verglichen. Wenn der neue Validierungsverlustwert um einen bestimmten Schwellenwert (Minimum Delta = 0,001) niedriger ist als der bisherige niedrigste Wert, wird dies als Fortschritt gewertet und der Zähler wird auf Null gesetzt. Wenn jedoch keine 5 aufeinanderfolgenden Entwicklungen festgestellt werden, erhöht sich der Zähler, und wenn diese Zahl die Schwelle überschreitet, wird das Training frühzeitig beendet. Dadurch werden Zeit und Ressourcen gespart, indem das Überlernen (Overfitting) des Modells verhindert und unnötige Epochen vermieden werden.

* 1. Semantisches Wörterbuch

Um diese Methode richtig zu applizieren, wurden die mehrkanaligen Merkmalskarten (C (Kanal), H (Höhe), W (Breite)) aus der letzten Konvolutionsschicht für jede Datei mit dem globalen Durchschnittspooling (GAP) für jeden Kanal reduziert. Diese Aktivierungen wurden mit den Gewichten der FC-Schicht für diese Klasse multipliziert, um den Beitrag jedes Kanals zur Klassenschätzung zu berechnen. Danach dieser Vorgang wurden für alle Kanalen, Klassen und Modelle wiederholt, um den durchschnittlichen Kanalbeitrags-Vektor (C,) für jede Klasse zu erhalten. Danach wurden die Kanäle mit dem höchsten fünf Beitrag in diesen Vektoren ausgewählt. Am Ende wurden extrahierte Kanäle dank der Grad-CAM Methode visualisiert. Allerdings wurde in dieser Studie im Gegensatz zur klassischen Grad-CAM-Anwendung nicht die gesamte Aktivierungslandkarte, sondern nur der Ausgang eines bestimmten Kanals visualisiert. Deswegen wurde verglichen, wie verschiedene ResNet-Modelle für denselben Kanal auf dieselben Daten aktiv werden.

Einfach zu sagen: Dieser Teil der Studie macht, dass die Extrahierung die Gewichten der Entscheidungsteil des Modells und kondensiert, und dann entscheidet, ob wichtig und wie wichtig die inneren unterschiedlichen Kanäle sind, und speichert die Information und Indizien der Kanäle.

* 1. Zurückschneiden und zweites Training

Bei diesem Teil der Studie wurden die gleichen gespeicherten Modelle ein zweites Mal nur mit gespeicherten Indizes mit den semantisch gespeicherten Kanälen trainiert. Um diese Methode richtig zu laufen, wurde die Zurückschneiden Methode appliziert, durch die Blockierung außer der Indizes im semantischen Wörterbuch (eigentlich semantisches List) der letzten Konvolutionsschicht. Jedes Modell mit jeder gespeicherten semantischen Indizien trainiert, mit drei Epoch Arten (10 Epoch, 25 Epoch und Frühstopp-Mekanismus). Also entstehen 27 verschiedene Modelle mit unterschiedlichen Scoren. Wie im ersten Training wurden alle Informationen über Modelle gespeichert.

1. Ergebnisse
   1. Ergebnisse der ersten Trainings

In der Tabelle 3.1 gibt es die standardisierten Trainingsskoren von insgesamt neun ResNet-Modellen als drei Trainingsstrategien. Bei dem Modell ResNet-34 wurde die höchste Genauigkeit bei Frühstop und 10 Epochen erreicht. Aber bei höheren Trainings-Epochen (25) wurde mit dem ResNet-50 Modell die höchste Genauigkeit erreicht. Diese Modelle wurden mit dem Fett-Style gezeichnet.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabelle 3.1:** Standard-Trainingsergebnis von Modellen. | | | | | |
| Epoche | Modell | Präzision | Erinnern | FI-Wertung | Genauigkeit |
| Frühstop  Mechanismus | ResNet18 | 0.9666 | 0.9641 | 0.9634 | 0.9641 |
| **ResNet34** | **0.9820** | **0.9818** | **0.9818** | **0.9818** |
| ResNet50 | 0.9794 | 0.9791 | 0.9791 | 0.9791 |
| 10 Epochen | ResNet18 | 0.9722 | 0.9716 | 0.9716 | 0.9716 |
| **ResNet34** | **0.9758** | **0.9759** | **0.9757** | **0.9759** |
| ResNet50 | 0.9713 | 0.9705 | 0.9706 | 0.9705 |
| 25 Epochen | ResNet18 | 0.9763 | 0.9759 | 0.9759 | 0.9759 |
| ResNet34 | 0.9752 | 0.9743 | 0.9739 | 0.9743 |
| **ResNet50** | **0.9803** | **0.9791** | **0.9793** | **0.9791** |

In der Abbildung 3.1 gibt es die Grafiken für Trainings- und Validierungsverlustkurven von insgesamt drei erfolgreichen ResNet-Modellen als drei Trainingsstrategien. Auch in der Abbildung 3.2 gibt es drei Grafiken für Verwirrungsmatrizen dieser drei Modelle.

|  |
| --- |
|  |
| **Abbildung 3.1:** Trainings- und Validierungsverlustkurven der Modelle als Trainingsstrategie. |

In der Abbildung 3.1 gibt drei Bilder für drei erfolgreichste Modelle zur der Beobachtung der Training- und Validierungsverlust Daten, in der Grafik *a)* gibt der Verlustkurve für ResNet34 mit der Frühstopp Strategie trainiert, in der Grafik *b)* gibt der Verlustkurve für ResNet34 bei 10 Epoch trainiert, in der Grafik *c)* gibt der Verlustkurve für ResNet50 bei 25 Epoch trainiert. X-Achse ist der Epoch zahlen und Y-Achse ist der Verlustwerten. Es ist beobachtet, dass in der Grafik *a)* und *b)* die Verlustwerte fast immer verringert, aber es ist nicht gleich bei Graphik *c)*. In der Graphik *c)* wurden die Verlustwerte Schwankungen gemacht.

|  |
| --- |
|  |
| **Abbildung 3.2:** Verwirrungsmatrizen der Modelle als Trainingsstrategie. |

In der Abbildung 3.2 gibt drei Bilder für drei erfolgreichste Modelle zur der Beobachtung der Verwirrungsmatrizen. In der Grafik *a)* gibt der Verwirrungsmatriz von ResNet34 mit der Frühstopp Strategie trainiert, in der Grafik *b)* gibt der Verwirrungsmatriz von ResNet34 bei 10 Epoch trainiert, in der Grafik *c)* gibt der Verwirrungsmatriz von ResNet50 bei 25 Epoch trainiert. X-Achse ist der vorhergesagte Klassen und Y-Achse ist der richtige Klassen.

* 1. Ergebnisse und Bildtransformation der semantischen Daten

Nach dem ersten Training, wurden semantische Daten aus den Modellen extrahiert, um die Kanäle zu sehen. Grad-CAM-Methoden wurden verwendet, um zu zeigen, wie Kanäle Bilder wahrnehmen und beeinflussen. Jeder ausgezeichnete Kanal wurde zurückgeschnitten. Es bedeutet, dass die letzte konvolutionale Schicht nur mit gezeichneten Kanälen beinhalten ist. Die Kanäle, die bei den verschiedenen Modellen mit unterschiedlichen Epoch-Zahlen gemeinsam gefunden wurden, wurden mit zufällig ausgewählten Bildern aus den verschiedenen Klassen zu einer Heatmap kombiniert und mit dieser Methode durch Überlappung visualisiert (Abbildung 3.3).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **renklilik içeren bir resim  Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.** | | | | |
| **Abbildung 3.3:** Eine Beispielabbildung der Ausgang von einen Kanal aus beiden ResNet18 und ResNet34 von 25 Epoch bei Klasse Bacterial Spot. | | | | |
| In der Abbildung 3.3 sind drei Heatmaps über ein Bacterial Spot Klass Bild überlagert. a) ResNet-18-Modell bei 25 Epochen und die Ausgang des Kanals 313, b) ResNet-34-Modell bei 25 Epochen und die Ausgang des Kanals 313, c) ResNet-50-Modell bei 25 Epochen und die Ausgang des Kanals 313. Rote seiten zeigen höhe Bedeutung und blaue Seiten zeigen geringere Bedeutung über der Bestimmung von Klassen. | | | | |
| **Tabelle 3.2:** Die gemeinsamen Kanäle und Zahlen aus drei Modellen von drei Trainingsstrategien. | | | | |
| Trainingsart | | Modelle | Gemeinsame Kanäle Anzahl | Gemeinsamer Kanäle Indizien |
| Frühstopp | | ResNet18-ResNet34 | 3 | 22, 97, 403 |
| ResNet18-ResNet50 | 1 | 56 |
| ResNet34-ResNet50 | 0 | - |
| 10 Epoch | | ResNet18-ResNet34 | 2 | 80, 136 |
| ResNet18-ResNet50 | 0 | - |
| ResNet34-ResNet50 | 1 | 234 |
| 25 Epoch | | ResNet18-ResNet34 | 5 | 141, 230, 257, 313, 372 |
| ResNet18-ResNet50 | 0 | - |
| ResNet34-ResNet50 | 0 | - |

In der Tabelle 3.2 gibt es gemeinsame Kanäle zwischen verschiedenen ResNet-Modellen. Grund der drei Modelle keine gemeinsamen Kanäle gehabt, wären entsprechende Zeilen nicht geschrieben worden. Es ist beobachtet, dass sowohl mit den meisten Kanälen pro Modell als auch mit den meisten Kanälen pro Trainingsstrategie die ResNet18-ResNet34-Modelle für 25 Epoch sehr ähnlich waren.

* 1. Ergebnisse des zweiten Trainings

Nach dem Zurückschneidungsprozess wurden die Modelle ein zweites Mal mit den bestimmten Kanälen Informationen trainiert. Die Scores zum Training der Modelle werden in der Tabelle 3.3 gezeigt. Auch in der Abbildung 3.4 und in der Abbildung 3.5 stehen die Grafiken zur Genauigkeit dieser Modelle.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabelle 3.3:** Zurückgeschnittene (pruned) Trainingsergebnisse von Modellen. | | | | | | |
| Trainingsstrategie | Kanälenherkunft | Modell | Präzision | Erinnern | FI-Wertung | Genauigkeit |
| Frühstop  Mechanismus | **ResNet18** | ResNet18 | 0.9860 | 0.9855 | 0.9856 | 0.9855 |
| ResNet34 | 0.9887 | 0.9882 | 0.9882 | 0.9882 |
| **ResNet50** | **0.9921** | **0.9920** | **0.9920** | **0.9920** |
| ResNet34 | ResNet18 | 0.9842 | 0.9839 | 0.9839 | 0.9839 |
| ResNet34 | 0.9829 | 0.9812 | 0.9815 | 0.9812 |
| ResNet50 | 0.9876 | 0.9871 | 0.9871 | 0.9871 |
| ResNet50 | ResNet18 | 0.9778 | 0.9769 | 0.9769 | 0.9769 |
| ResNet34 | 0.9840 | 0.9834 | 0.9834 | 0.9834 |
| ResNet50 | 0.9740 | 0.9700 | 0.9708 | 0.9700 |
| 10 Epoch | **ResNet18** | ResNet18 | 0.9803 | 0.9802 | 0.9801 | 0.9802 |
| **ResNet34** | **0.9899** | **0.9898** | **0.9898** | **0.9898** |
| ResNet50 | 0.9851 | 0.9850 | 0.9850 | 0.9850 |
| ResNet34 | ResNet18 | 0.9645 | 0.9630 | 0.9632 | 0.9630 |
| ResNet34 | 0.9817 | 0.9812 | 0.9813 | 0.9812 |
| ResNet50 | 0.9789 | 0.9775 | 0.9776 | 0.9775 |
| ResNet50 | ResNet18 | 0.9823 | 0.9823 | 0.9822 | 0.9823 |
| ResNet34 | 0.9835 | 0.9834 | 0.9834 | 0.9834 |
| ResNet50 | 0.9752 | 0.9743 | 0.9737 | 0.9743 |
| 25 Epoch | **ResNet18** | ResNet18 | 0.9762 | 0.9727 | 0.9719 | 0.9727 |
| **ResNet34** | **0.9968** | **0.9968** | **0.9968** | **0.9968** |
| ResNet50 | 0.9825 | 0.9812 | 0.9813 | 0.9812 |
| ResNet34 | ResNet18 | 0.9937 | 0.9936 | 0.9936 | 0.9936 |
| ResNet34 | 0.9918 | 0.9914 | 0.9915 | 0.9914 |
| ResNet50 | 0.9838 | 0.9834 | 0.9832 | 0.9834 |
| ResNet50 | ResNet18 | 0.9952 | 0.9952 | 0.9952 | 0.9952 |
| ResNet34 | 0.9904 | 0.9903 | 0.9903 | 0.9903 |
| ResNet50 | 0.9942 | 0.9941 | 0.9941 | 0.9941 |

In der Tabelle 3.3 gibt es die Endskoren für 27 Modelle bei drei unterschiedlichen Trainingsstrategien und für jedes vorhertrainierte Modell extrahierte Kanalinformationen. In der ersten Spalte sind die Trainingsstrategien; in der zweiten Spalte stehen die Informationen der extrahierten Kanäle aus vorhertrainierten Modellen; In der dritten Spalte die Informationen der zum zweiten Mal trainierten Modelle mit spezifischen Kanälen; bei der vierten, fünften, sechsten und siebten Spalte gibt es bestimmte Werte für die Modelle. In der zweiten Spalte gibt es drei fettgeschriebene Modelle. Sie sind die gleichen Modelle in der Tabelle 3.1 mit dem höchsten Genauigkeitswert als drei Trainingsstrategien. In der dritten Spalte gibt es 3 fettgezeichnete Modelle für jede Trainingsart und bestimmte Kanäle.

Um leichter zu beobachten, wurden in der Graphik in Abbildung 3.4 die Genauigkeitswerte von Modellen als die Trainingsstrategien gruppiert und visualisiert.

|  |
| --- |
|  |
| **Abbildung 3.4**: Genauigkeitsgraph von Modellen nach Zurückschneidung. |

In der Abbildung 3.4 wurden die Genauigkeitswerten für alle Modelle von verschiedenen Kanalherkünften und Trainingsstrategien visualisiert. Um leichter zu sehen, wurden die höchsten 3 Modelle als Trainingsstrategien angenommen und in der Abbildung 3.5 gezeichnet.

|  |
| --- |
|  |
| **Abbildung 3.5:** Höchste Genaugkeitswerte nach Zurückschneidung. |

In der Abbildung 3.5 wurden die höchste genauigkeithabende Modelle und ihren Genauigkeitswerte als Trainingsstrategie dargestellt. Es wurde beobachtet, dass das ResNet34-Modell, das in 25 Epochen trainiert wurde, einen Unterschied von 0,007 und 0,005 gegenüber den anderen beiden Modellen erzielte. Wegen ihres Erfolges bei ihrer Trainingsart wurden die Verlustgraphiken dieser drei Modelle in der Abbildung 3.6 gezeichnet.

|  |
| --- |
|  |
| **Abbildung 3.6**:Trainings- und Validierungsverlustkurven für 3 erfolgreichste Modelle als ihre Trainingsstrategien. |

In der Abbildung 3.6 stehen drei Verlustgrafiken für die drei erfolgreichsten Modelle als Trainingsstrategie. *a)* ist die Verlustgrafik für ResNet-50-Modell mit Frühstopp-Mekanismus und mit den Kanälen aus mit Frühstopp trainiertem ResNet18 Modell; *b)* ist die Verlustgrafik für ResNet-34-Modell mit 10 Epoch trainiert und mit den Kanälen aus mit 10 Epoch trainiertem ResNet18 Modell; *c)* ist die Verlustgrafik für ResNet-34-Modell mit 25 Epoch trainiert und mit den Kanälen aus mit 25 Epoch trainiertem ResNet18-Modell.

Am Ende der Trainings wurde beobachtet, dass eine Erhöhung des Mittelwerts für jedes Modell zwischen 0.002 und 0,02 lag. Es passierte auch, dass die Trainingszeit der Modelle durchschnittlich um 25–30 % verkürzt wurde.

1. Diskussion
   1. Diskussion
      1. Dateianzahlen

Dateienzahl kann ein großes Vorurteilengrund sein, weil es sehr viel mehr Tomato Yellow Leaf Curl Virus-Dateien gibt. Es lässt sich sehr leicht kommentieren, dass die Modelle dieser Klas mehr als die anderen zu erinnern lernen. Die Zahlen für andere Datenklassen sind ungefähr ähnlich, aber die nächste Datei nach der Tomato Yellow Leaf Curl Virus Klasse ist mit ungefähr 3200 Dateien weniger Bacterial Spot Klas. Es ist eine große Dateimenge für einen Datensatz mit 18335 Dateien.

* + 1. Trainingsfehlern

Wie in Abbildung c in Abbildung 3.6 ersichtlich ist, treten die Schwankungen in den Modellen ungefähr in der 15. Epoche auf, in einigen Modellen sogar schon ab der 10. Epoche. Es treten bei den Versionen, die 25 Epochen lang trainiert werden, Modelle auf, anstatt bei den Versionen, die mit Frühstopp oder 10 Epochen trainiert werden. Deswegen kann man annehmen, dass die Modelle, die 25 Epochen lang trainiert wurden, möglicherweise überangepasst waren.

Diese Modelle wurden 25 lang ohne Anwendung einer Zurückschneidung (Pruning) trainiert, und also vor dem zweiten Mal mit Zurückschneidungs-Trainierung zeigen die Modelle bei 25 Epoch die Schwankung auch. Die Modelle wurden 25 Epochen lang ohne Anwendung einer Zurückschneidenmethode trainiert. Wenn sie jedoch mit klassenspezifisch bestimmten Kanälen ein zweites Mal trainiert werden, treten diese Schwankungen auf. Dies kann als Indikator dafür angesehen werden, dass die Modelle nun das Lernen eingestellt haben. Denn in dieser Situation können die Modelle beginnen, die zuvor korrekt eingestellten Gewichts- und Verschiebungswerte zu ändern.

Somit kann man Folgendes kommentieren: Mit zunehmender Anzahl der Epochen steigt auch das Risiko einer übermäßigen Überlernung. Also, es ist auch kommentierbar, dass die optimierte Epoch-Zahl für diesen Datensatz 10 zu 25 ist. Es ist ersichtlich bei der Abbildung 3.1.

Es ist auch kommentierbar, dass die 5 der total (ungefähr 50) Kanäle aus dem ResNet-34-Modell bei 25 Epochen Epoch gemeinsam sind. Deswegen hat bei der Trainierung des Modells ResNet-34 bei 25 Epoch aus der Kanäleherkunft ResNet-18-Modells schnell überlernt. Wenn es keine gemeinsamen Kanäle zwischen ResNet-18- und ResNet-34-Modellen gibt, werden die Überlernungen weniger oder bei weiteren Epochzahlen entstehen. Weil die zum zweiten Mal trainierten Modelle schon einmal trainiert wurden, je mehr die Epochenzahl stieg, desto mehr die Gefahr der Überlernung.

Aber diese Anpassungen zeigen, dass diese Modelle anfangen, die Daten einzuprägen. Dennoch hat es eine hohe Genauigkeit und es wird nutzbar sein, wenn die zurückgeschnittenen Epoch Zahlen optimiert werden.

* + 1. Vergleich zwischen erwarteten und angenommenen Ergebnissen

Für diese Studie wurde erwartet, dass das mit Frühstopp trainierte ResNet-34-Modell unter Verwendung der aus ebenfalls mit Frühstopp trainierten ResNet-34-Modellen gewonnenen Kanalinformationen die höchste Genauigkeit erreicht. Stattdessen wurden bei der Verwendung von kreuzweise genutzten Kanalinformationen in allen drei Trainingsstrategien die besten Ergebnisse erzielt, bei 10 und 25 Epochen für ResNet18–ResNet34 sowie beim Frühstopp für ResNet18-ResNet50. Dieses Ergebnis ist zunächst unerwartet, erweist sich jedoch bei näherer Betrachtung als plausibel. Wie bereits zuvor erläutert wurde, steigt die Wahrscheinlichkeit für Überlernung, je länger ein Modell mit seinen eigenen Merkmalen (Kanälen) trainiert wird. Detaillierte Informationen sind in Tabelle 4.1 dargestellt.

Wie bereits zuvor erläutert wurde, zeigte das ResNet18-ResNet34-Modell sowohl bei 10 als auch bei 25 Epochen im Vergleich zu den anderen konzipierten Modellen eine höhere Leistung, begann jedoch bei höheren Epochenzahlen Anzeichen des Überlernens zu zeigen. Es wird dennoch angenommen, dass ein ResNet18-ResNet34-Modell, dessen Training vor dem Einsetzen von Überlernung gestoppt wird, im Rahmen der Zurückschneidenmethode eine deutlich höhere Erfolgsquote erzielen könnte.

Denn Selbst in seinem Überlernungsepassten Zustand weist dieses Modell sowohl die höchste Genauigkeit als auch die größte Veränderung in der Genauigkeit auf. Wie bereits zuvor erwähnt, ist auch die ausgeprägte Schwankung im Verlustsverlauf ein Aspekt, der ausführlich diskutiert werden kann. Im Durchschnitt erreichte dieses Modell in allen drei Trainingsstrategien den höchsten mittleren Genauigkeitswert.

Bei anderen Modellen hingegen wurde (sowohl aufgrund von Überlernung als auch durch die Auswahl irrelevanter oder ungeeigneter Kanäle) statt einer Leistungssteigerung sogar ein Rückgang der Genauigkeit zwischen -0.0006 und -0.0091 beobachtet. Dennoch lässt sich im Vergleich zwischen der Steigerungs- und der Abnahmerate der Modellgenauigkeit feststellen, dass diese Methode insgesamt zur Leistungsverbesserung der Modelle beiträgt.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabelle 4.1:** Genauigkeitsänderungen, Genauigkeitswerte und Genauigkeitsmittelwerte für 3 ResNet-Modelle mit 3 unterschiedlichen Kanäleherkunften. | | | | | | | |
| **Kanäleherkunft - Modelle** | **Genauigkeitsänderungen** | | | **Genauigkeitswerte** | | | **Genauigkeits Mittelwerte** |
| **Frühstopp** | **10 Epoch** | **25 Epoch** | **Frühstopp** | **10 Epoch** | **25 Epoch** |
| RN18-RN18 | **0,0214** | 0,0086 | -0,0032 | 0,9855 | 0,9802 | 0,9727 | 0,9794 |
| **RN18-RN34** | 0,0064 | 0,0139 | **0,0225** | 0,9882 | **0,9898** | **0,9968** | **0,9916** |
| **RN18-RN50** | 0,0129 | **0,0145** | 0,0021 | **0,9920** | 0,9850 | 0,9812 | 0,9860 |
| RN34-RN18 | 0,0198 | -0,0086 | 0,0177 | 0,9839 | 0,9630 | 0,9936 | 0,9801 |
| RN34-RN34 | -0,0006 | 0,0053 | 0,0171 | 0,9812 | 0,9812 | 0,9914 | 0,9846 |
| RN34-RN50 | 0,0080 | 0,0070 | 0,0043 | 0,9871 | 0,9775 | 0,9834 | 0,9826 |
| RN50-RN18 | 0,0128 | 0,0107 | 0,0193 | 0,9769 | 0,9823 | 0,9952 | 0,9848 |
| RN50-RN34 | 0,0016 | 0,0075 | 0,0160 | 0,9834 | 0,9834 | 0,9903 | 0,9857 |
| RN50-RN50 | -0,0091 | 0,0038 | 0,0150 | 0,9700 | 0,9743 | 0,9941 | 0,9794 |

In der Tabelle 4.1 stellt die Modellengenauigkeitsänderungen zwischen erste und zweite Trainings und Genauigkeitswerten für Modellen (gleich Daten im Tabelle 3.3), in der letzte Spalte stellt die Genauigkeitsmittelwerten von drei Trainingsstrategie für jeder Kanäleherkunft-Model paar.

* + 1. Verglichen mit Literatur

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabelle 4.2:** Tabelle zur Literaturvergleich. | | | | |
| **Von** | **Modell** | **Datensatz** | **Method** | **Genauigkeit** |
| [21] | CNN, ResNet50, Efficient B3, Generic CNN | PlantVillage | Transfer Lernen | CNN: 99.68%  ResNet50: -  Efficient B3: -  Generic CNN: - |
| [22] | ResNet-34-basierte Faster-RCNN | PlantVillage | ROI Annotation, CBAM-basiertes Merksmal  Extrahierung | 99.97% |
| [23] | Generative Adversarial Classified Network (GACN) | PlantVillage  AI Challenger 2018 | GAN zur Datensatz-Stabilisierung | AI Challenger: 89.23%, PlantVillage: 99.43% |
| [10] | Inception V3 | PlantVillage | Visualization, Feature Extraction  attention maps | 97.4% |
| [13] | tiefe CNN (Caffe Framework) | Eigenen Datensatz mit 13 Klassen | Tiefes Merksmal  Extrahierung | Durchschnittlich  96.3% (91-98%) |
| [16] | ML Bildverarbeitung | ESCA-Datensatz | Bildverarbeitung | - |
| [15] | CNN (eingebettet) | ESCA,  PlantVillage(PV) | filterschnitt | ESCA: 98.10%  PV: 95.24% |
| [17] | Enhanced ANN und CNN | Eigenen Datensatz | Segmentierung  (Enhanced ANN), Klassifizierung  (CNN) | Durchschnitt  93.75% |
| [14] | VGG-16, VGG-19, ResNet, Inception V3 | Labor- und feldliche Datensätze | Merksmal  Extrahierung, Transferlernen | Inception V3:  Labordaten: 99.35% feldlich: 84-89% |
| [24] | SVM | PlantVillage - Tomatendaten | SIFT-Merkmal  Extrahierung  GEV Modell | Quadratic SVM:  84.7% |
| [25] | LeNet basierte CNN | PlantVillage - Tomatendaten | automatisierte CNN-Merkmale  Extrahierung  Daten Augmentierung | Durchschnitt  94-95%  (der Beste 94.8%) |
| **Dieser Studie** | **ResNet-18, ResNet-34, ResNet-50** | **PlantVillage - Tomatendaten** | **Kanäle Extrahierung, Wärmekarten**  **Vergleich,**  **Zurückschneiden, drei Trainingsstrategien** | **ResNet18 extrahierte Kanäle bei 25 Epochen über ResNet34: 99.68%**  **bei 10 Epoch: 98.98%;**  **ResNet18 extrahierte Kanäle bei Frühstopp über ResNet50: 99.20%** |

* 1. Zukünftige Arbeiten
     1. Modelle verbessern

Bei dieser Studie wurden unterschiedliche Batchgrößen probiert, aber es ist nötig, um bessere Modelle zu erhalten, mehrere Male und unterschiedliche Hyperparameter zu probieren. Also nur für Batchgrößen wurden bei dieser Studie die möglichst schnellsten und auch mit die höchsten Genauigkeiten und besten Verlustgrafiken [26].

Die Verbesserungen bei den Epoch Zahlen werden auch eine gute Optimierungsmethode. Wie vorher gezeigt: Wenn die Epochenzahlen bei der zweiten Trainingsstufe höher sind, werden die Modelle mehr überlernen. Also: Weniger Stufen bei gleichen Hyperparametern oder generell optimierte Hyperparameter mit optimierten Epochzahlen werden Modelle besser sein.

* + 1. Automatisierung mit anderen Systemen und Applizieren im Feld

Nach der Modellierung können die Modelle exportiert werden und werden nutzbar bei der der Automatisierung bei verschiedenen Systemen. Als Beispiel: In einer Studie wurde durch die Kombination verschiedener Architekturen des maschinellen Lernens ein ganzheitliches System entwickelt, das von Anfang bis Ende funktioniert. Der Prozess beginnt mit der Erkennung von Blättern in den landwirtschaftlichen Bildern, die mit Drohnen aufgenommen wurden, und geht dann mit der Entfernung des Hintergrunds der Krankheitsklassifizierung, der Schadensegmentierung, der Schadenanalyse und der Behandlungsempfehlung weiter. Bei jeder Stufe wurde das am besten geeignete tiefes Lernen Modell für die jeweilige Aufgabe verwendet. Genutzte Modelle sind: YOLOv8 Modell für die schnelle und genaue Objekterkennung, CNN für die Klassifizierung und die UNet-Architektur für detaillierte Flächenauswertungen verwendet. So kann das System Krankheiten auf dem Bild erkennen und beschädigte Bereiche auf den Blättern pixelgenau unterscheiden [27].

1. literaturverzeichnis

[1] Habib-Ur-Rahman, M., Ahmad, A., Raza, A., Hasnain, M. U., Alharby, H. F., Alzahrani, Y. M., Bamagoos, A. A., Hakeem, K. R., Ahmad, S., Nasim, W., Ali, S., Mansour, F., & Sabagh, A. E. (2022). Impact of climate change on agricultural production; Issues, challenges, and opportunities in Asia. Frontiers in Plant Science, 13. https://doi.org/10.3389/fpls.2022.925548

[2] TÜİK - Veri portalı. (n.d.). https://data.tuik.gov.tr/Kategori/GetKategori?p=tarim-111&dil=1

[3] Hughes, D. P., & Salathe, M. (2015, November 25). An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv.org. https://arxiv.org/abs/1511.08060

[4] Alpaydin, E. (2019). Maschinelles Lernen. In De Gruyter eBooks. https://doi.org/10.1515/9783110617894

[5] Amaratunga, T. (2021). Deep learning on Windows: Building Deep Learning Computer Vision Systems on Microsoft Windows. Apress.

[6] LeCun, Y., Bottou L., Bengio Y., & Haffner P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324. https://doi.org/10.1109/5.726791

[7] Tan, P., Steinbach, M., & Kumar, V. (2005). Introduction to Data Mining, (First Edition). In Addison-Wesley Longman Publishing Co., Inc. eBooks. https://dl.acm.org/citation.cfm?id=1095618

[8] He, K., Zhang, X., Ren, S., & Sun, J. (2015, December 10). Deep residual learning for image recognition. arXiv.org. https://arxiv.org/abs/1512.03385

[9] And, N. a. A., & Prince. (2021). A classification of Arab ethnicity based on face image using deep learning approach. IEEE Access, 9, 50755–50766. https://doi.org/10.1109/access.2021.3069022

[10] Toda, Y., & Okura, F. (2019). How convolutional neural networks diagnose plant Disease. Plant Phenomics, 2019. https://doi.org/10.34133/2019/9237136

[11] Li, H., Kadav, A., Durdanovic, I., Samet, H., & Graf, H. P. (2016, August 31). Pruning filters for efficient ConVNets. arXiv.org. https://arxiv.org/abs/1608.08710

[12] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2019). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. International Journal of Computer Vision, 128(2), 336–359. https://doi.org/10.1007/s11263-019-01228-7

[13] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. Computational Intelligence and Neuroscience, 2016, 1–11. https://doi.org/10.1155/2016/3289801

[14] Ahmad, I., Hamid, M., Yousaf, S., Shah, S. T., & Ahmad, M. O. (2020). Optimizing pretrained convolutional neural networks for tomato leaf disease detection. Complexity, 2020, 1–6. https://doi.org/10.1155/2020/8812019

[15] Falaschetti, L., Manoni, L., Di Leo, D., Pau, D., Tomaselli, V., & Turchetti, C. (2022). A CNN-based image detector for plant leaf diseases classification. HardwareX, 12, e00363. https://doi.org/10.1016/j.ohx.2022.e00363

[16] Alessandrini, M., Calero Fuentes Rivera, R., Falaschetti, L., Pau, D., Tomaselli, V., & Turchetti, C. (2021). A grapevine leaves dataset for early detection and classification of esca disease in vineyards through machine learning. Data in brief, 35, 106809. https://doi.org/10.1016/j.dib.2021.106809

[17] Chen, J., Zhang, D., Nanehkaran, Y. A., & Sun, Y. (2021). A cognitive vision method for the detection of plant disease images. Machine Vision and Applications, 32(1). https://doi.org/10.1007/s00138-020-01150-w

[18] G, G., & J, A. P. (2019). Identification of plant leaf diseases using a nine-layer deep convolutional neural network. Computers & Electrical Engineering, 76, 323–338. https://doi.org/10.1016/j.compeleceng.2019.04.011

[19] J, A. P., & Gopal, G. (2019). Data for: Identification of Plant Leaf Diseases Using a 9-layer Deep Convolutional Neural Network. -, 1. https://doi.org/10.17632/tywbtsjrjv.1

[20] Deng, J., Dong, W., Socher, R., Li, L., Li, N. K., & Fei-Fei, N. L. (2009). ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/cvpr.2009.5206848

[21] Sharma, A., Bansal, R., & Bansal, T. (2025). Transforming Agriculture: Plant Disease Detection with Transfer Learning and Deep Neural Network. In Advances in intelligent systems research/Advances in Intelligent Systems Research (pp. 482–495). https://doi.org/10.2991/978-94-6463-716-8\_38

[22] Nawaz, M., Nazir, T., Javed, A., Masood, M., Rashid, J., Kim, J., & Hussain, A. (2022). A robust deep learning approach for tomato plant leaf disease localization and classification. Scientific Reports, 12(1). https://doi.org/10.1038/s41598-022-21498-5

[23] Wang, X., & Cao, W. (2023). GACN: Generative Adversarial Classified Network for Balancing Plant Disease Dataset and Plant Disease Recognition. Sensors, 23(15), 6844. https://doi.org/10.3390/s23156844

[24] Model-Based statistical features for mobile phone Image of tomato plant disease classification. (n.d.). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/8327092

[25] Tomato leaf disease detection using convolutional neural networks. (n.d.). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/8530532

[26] Keskar, N. S., & Socher, R. (2017, December 20). Improving Generalization Performance by Switching from Adam to SGD. arXiv.org. https://arxiv.org/abs/1712.07628

[27] Polly, R., & Devi, E. A. (2024). “Semantic segmentation for plant leaf disease classification and damage detection: A deep learning approach.” Smart Agricultural Technology, 9, 100526. https://doi.org/10.1016/j.atech.2024.100526