

Überblick

Aus verschiedenen Gründen kann es von Interesse sein, bestehende Leiterplatten und die darauf auffindbaren Schaltungen zu analysieren.

Dies kann verschiedene Ziele haben, zum Beispiel:

- Analyse eines defekten Gerätes, um die Suche nach defekten Komponenten zu erleichtern/zu ermöglichen oder defekte Teilschaltungen in einem Gesamtgerät zu ersetzen
- Analyse von Schaltungen anderer Firmen, um die Funktionsweise dieser Schaltungen zu verstehen und zum Beispiel auf Patentverletzungen zu überprüfen

Solche Analyse von Leiterplatten erfordert jedoch ein erhebliches Zeitinvestment und tiefgreifendes Wissen.

Wir zeigen 2 Teile eines Tools, welches den Arbeitsaufwand zur Umwandlung von Fotos 2-lagiger Leiterplatten in Schaltpläne stark verringert und erleichtert.

Datensatz

Die Betrachtung von PCB im Zusammenhang mit Machine Learning ist ein selten betrachtetes Thema. Entsprechend existieren nur etwa eine Handvoll Datensätze mit unterschiedlichen Zwecken, wie Labeling von Komponenten oder Klassifikation von Defekten. Unser trainiertes Neuronales Netz stützt sich auf den FPIC-Datensatz (Ref. FPIC). Dieser weist leider ein unvollständiges Labeling auf, welches später überarbeitet werden musste (siehe Abb. 1).

	Original	Bearbeitet		Original	Bearbeitet
R	10533	14055	FB	104	224
C	9617	12999	CRA	54	0
U	986	0	SW	52	71
Q	821	0	T	58	86
J	704	1510	F	69	94
L	710	725	V	41	8
RA	438	0	LED	45	295
D	461	704	S	39	33
RN	374	880	QA	37	32
TP	282	250	JP	147	39
IC	300	3158	CN	205	60
CR	255	5	Y	89	140
M	79	0	VR	18	6
BTN	72	76			

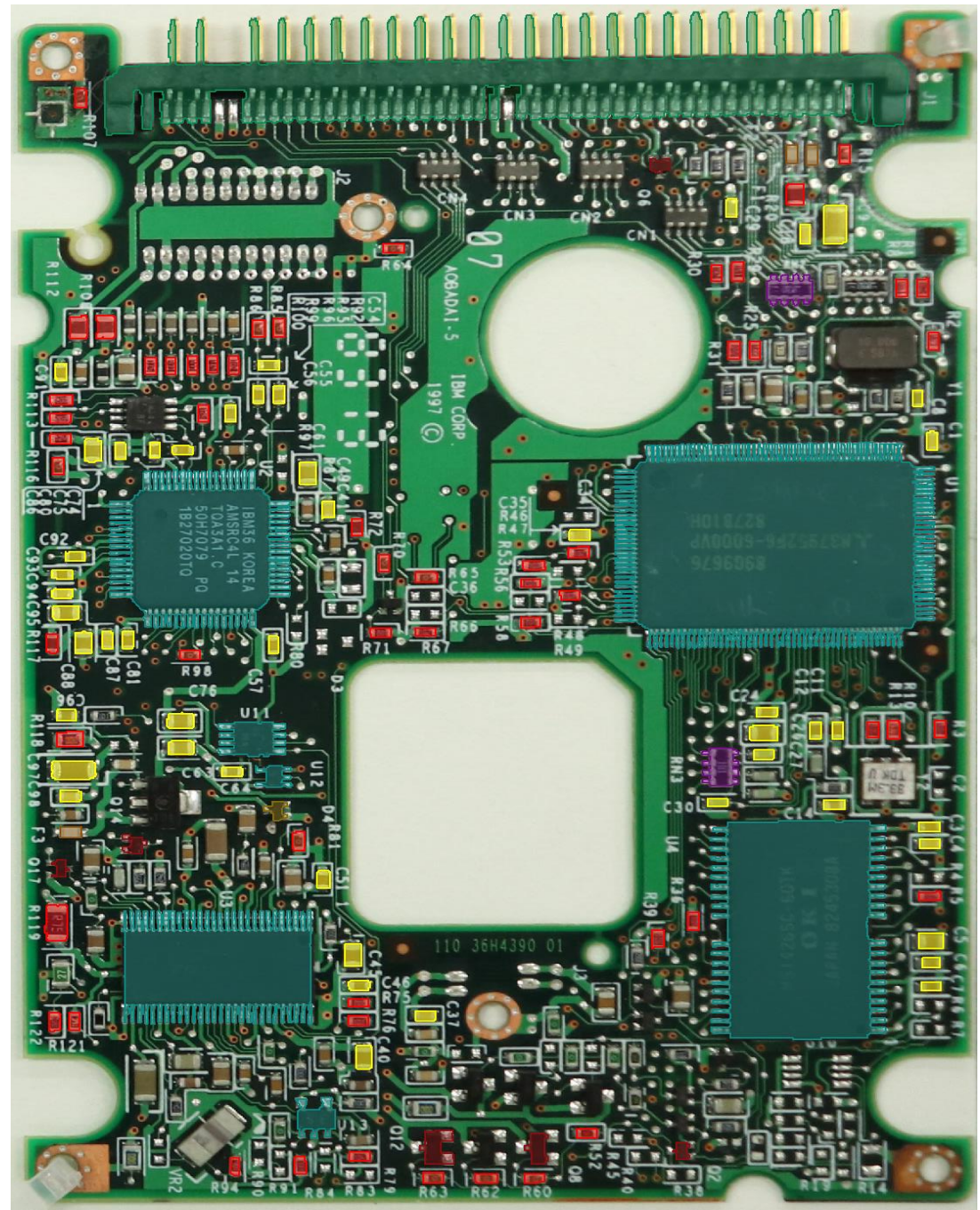
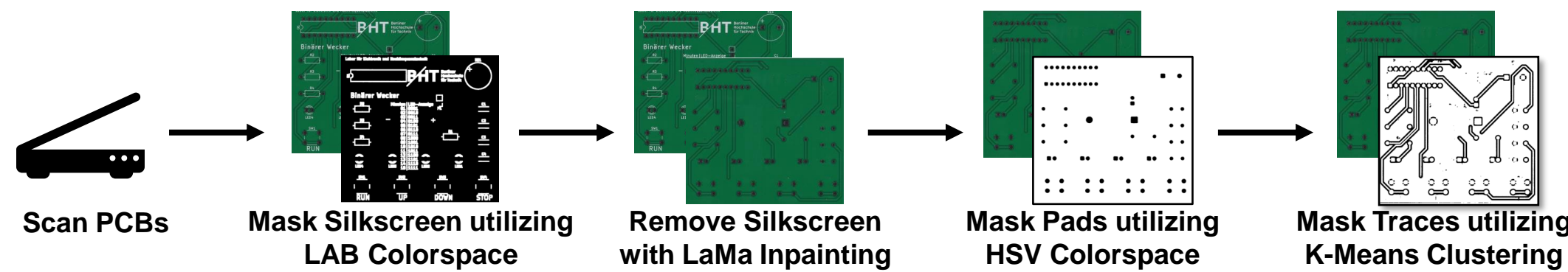


Abb. 1: Platine einer HDD mit unvollständig gelabelten Komponenten aus dem FPIC-Datensatz

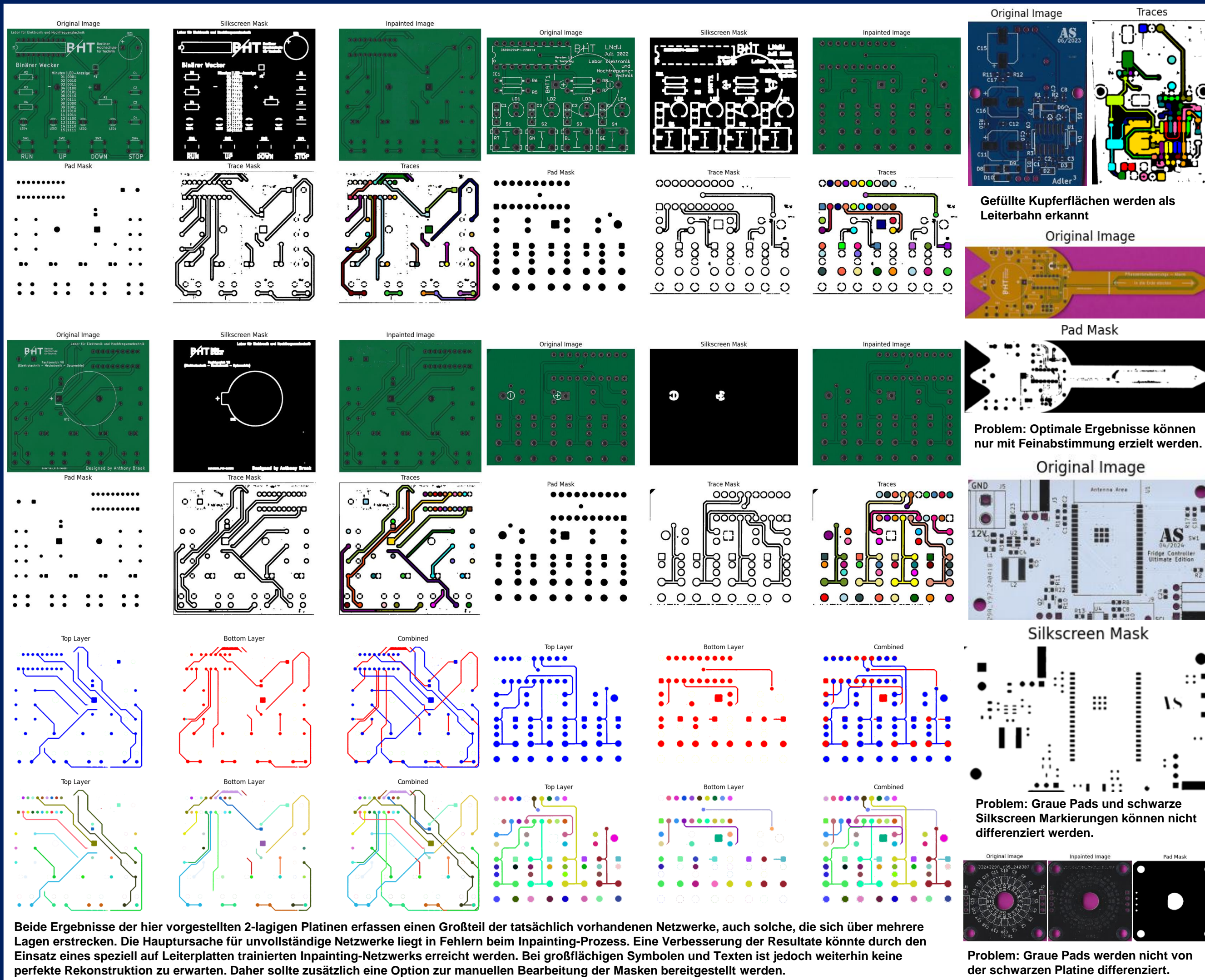
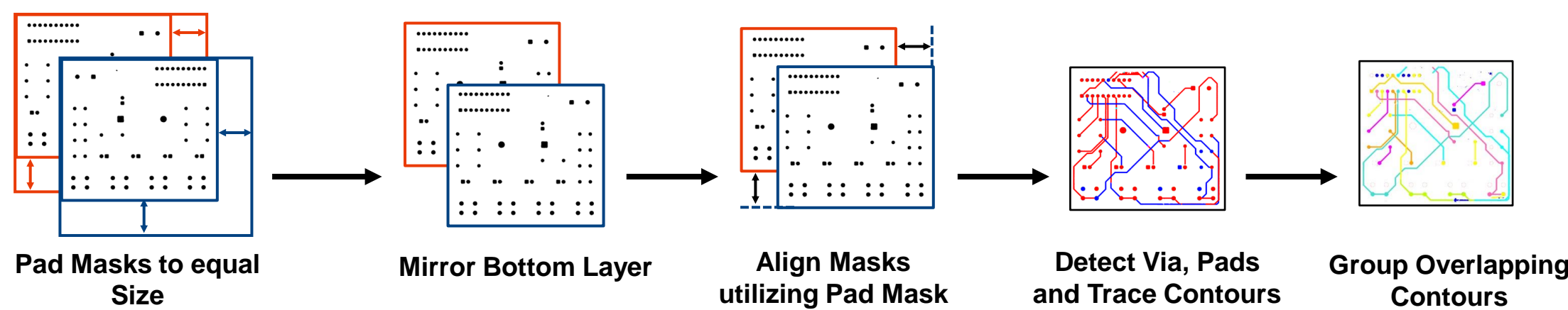
Trace Detection

Bisherige Ansätze zur Leiterbahnerkennung haben sich auf 1-lagige Platinen, ohne Silkscreen Markierungen und gefüllte Kupferflächen konzentriert (Ref. Reverse Engineering of Printed Circuit Boards). Nachfolgend wird die Analyse 2-lagiger Platinen mit Silkscreen Markierungen und gefüllten Kupferflächen vorgestellt.

Für die reproduzierbare Aufnahme von Leiterplatten Bildern wurde ein regulärer Flachbettscanner verwendet. Pro Platine wurden jeweils Vor- sowie Rückseite eingescannt. Die lagenweise Extraktion von Vias, Pads und Traces konnte mittels eines mehrstufigen Maskierungsprozess, unter Zuhilfenahme eines vortrainierten LaMa Inpainting Models, erreicht werden.



Die Analyse mehrlagiger Platinen erfordert eine lagen konforme Spiegelung sowie Ausrichtung der zuvor erzeugten Masken, welche erst nach Anpassung der Maskengrößen möglich ist. Mittels hierarchischer Konturerkennung können, aus den Pad und Trace Masken, Pads, Vias und Traces bestimmt werden. Jede zuvor erkannte Kontur wird Initial einem eigenen Netz zugewiesen. Für die Lagen unabhängige Netzwerk Erkennung wird die Überlappung der Konturen verglichen und verbundene Netzte zusammengefasst.

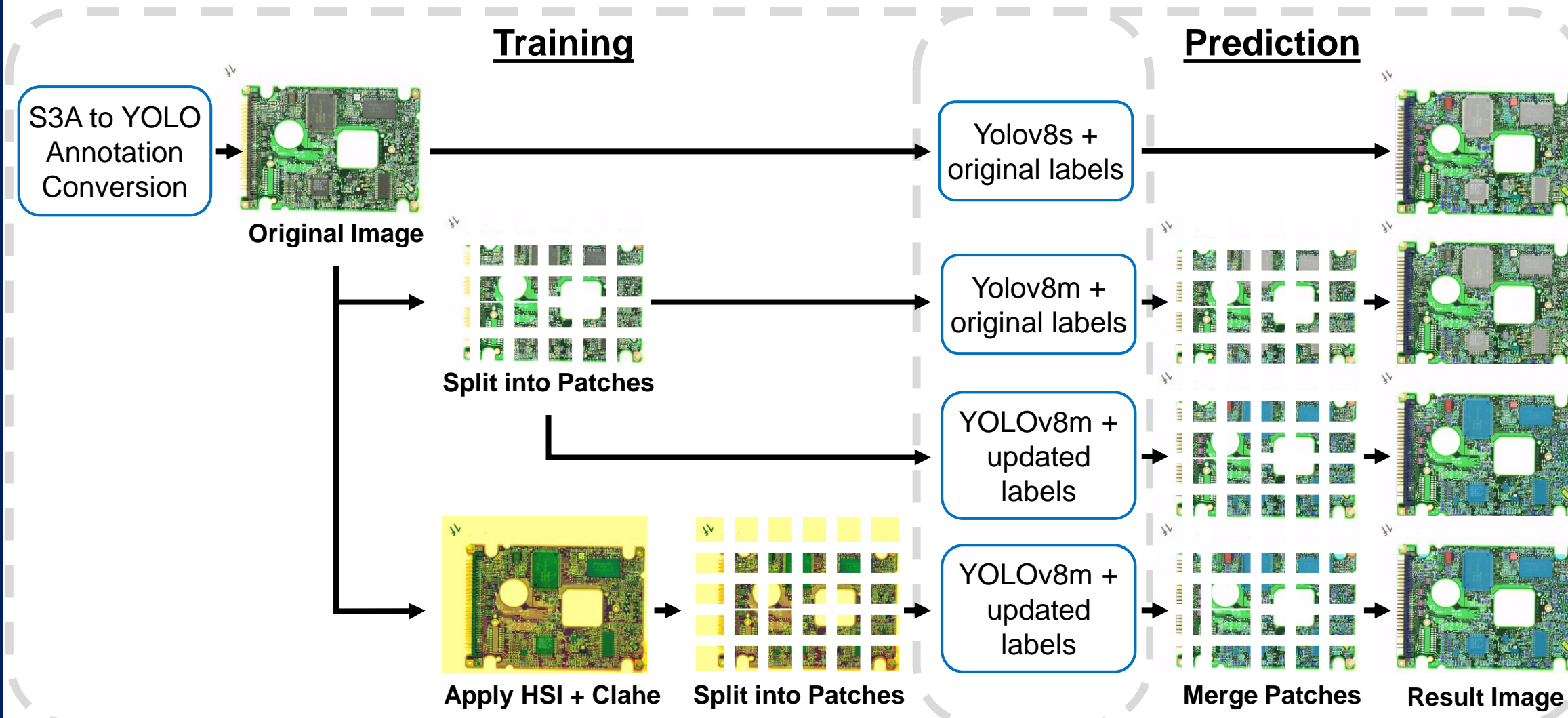


Beide Ergebnisse der hier vorgestellten 2-lagigen Platinen erfassen einen Großteil der tatsächlich vorhandenen Netzwerke, auch solche, die sich über mehrere Lagen erstrecken. Die Hauptursache für unvollständige Netzwerke liegt in Fehlern beim Inpainting-Prozess. Eine Verbesserung der Resultate könnte durch den Einsatz eines speziell auf Leiterplatten trainierten Inpainting-Netzwerks erreicht werden. Bei großflächigen Symbolen und Texten ist jedoch weiterhin keine perfekte Rekonstruktion zu erwarten. Daher sollte zusätzlich eine Option zur manuellen Bearbeitung der Masken bereitgestellt werden.

Problem: Graue Pads und schwarze Silkscreen Markierungen können nicht differenziert werden.

Problem: Graue Pads werden nicht von der schwarzen Platine differenziert.

Component Detection



Als Basis für das neuronale Netz wurde sich für das YOLOv8 Framework entschieden. Ziel war es, für verschiedenste Platinen eine möglichst gute Segmentierung und Klassifikation aller Komponenten zu erreichen. Hierzu konvertierten wir die gegebenen Annotationen des Datensatzes in das YOLO-Format und trainierten 4 verschiedene Netze mit iterativen Veränderungen.

- Ein Netz mit dem unveränderten Datensatz nach Konvertierung
- Aufteilen der Bilder in „flicker“ und damit Training mit einheitlich großen Bildern (ähnlich Ref. „PCBSegClassNet“)
- Eine Überarbeitung der Annotationen, um Klassen zu vereinheitlichen, fehlende Klassifikationen hinzuzufügen und einige Fehler zu korrigieren z.B. Q, U, MSP(IC-Name), IC zu IC vereinheitlicht / LD und LED zu LED / Y und X zu Y (Quarz) / Korrektur einiger Quarze von Q zu Y
- Überführung der Bilder in den HSI-Farbraum mit nachgelagerter Histogram Equalization (CLAHE) und anschließendem Training mit den überarbeiteten Annotationen (ähnlich Ref. „PCBSegClassNet“)

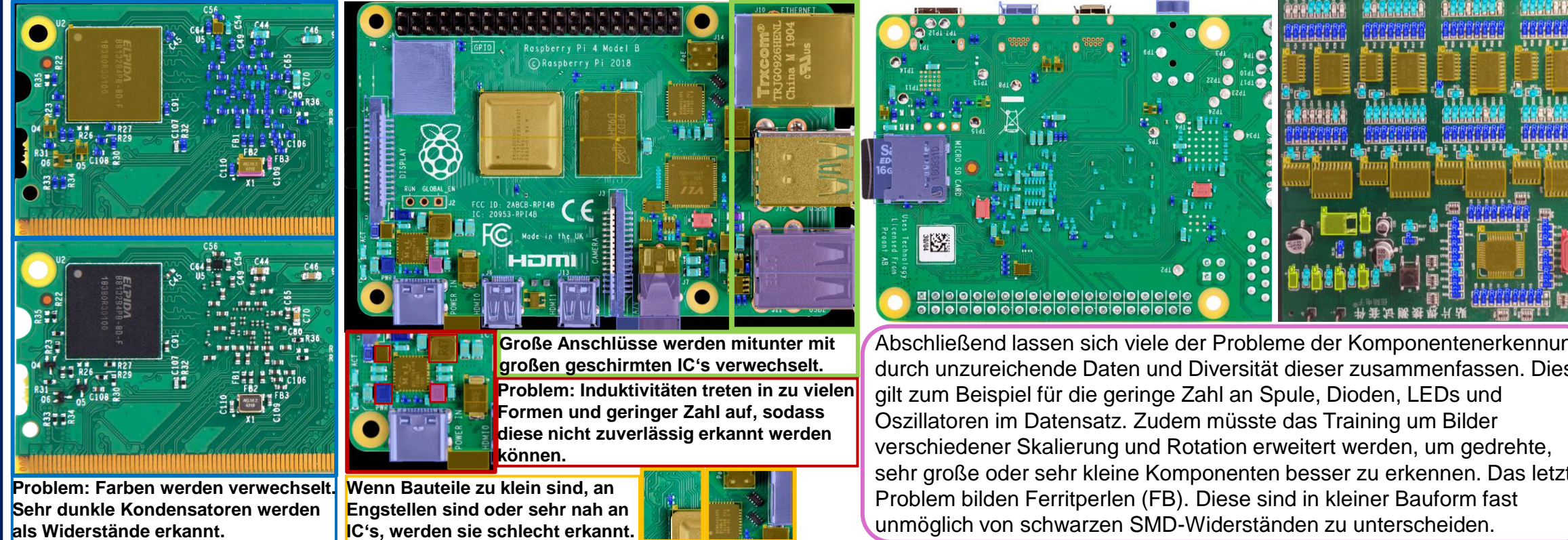
Anzahl markierter Komponenten je Netz für verschiedene Beispiel-PCBs

Ordner	# Bilder	Netz 1	Netz 2	Netz 3	Netz 4	Gesamt
arduino	15	347	340	723	395	941
Eigene Bilder	17	144	154	219	201	328
→ davon THT	10	63	84	115	91	181
raspberry	10	349	626	872	486	1139
smd	5	516	880	901	487	1227
Gesamt		1419	2084	2830	1660	3816

Anteilig	37.1%	54.6%	74.1%	43.5%
Nur THT	34.8%	46.4%	63.5%	50.3%

Auswertung der True-Positiv Rate der am meisten vorkommenden Klassen

Ordner	Bilder	IC	C	R	Y	LED	D	FB	L	RN
arduino	8	47	45	141	129	96	89	5	2	14
Eigene Bilder	17	39	38	34	31	89	85	0	0	4
tht-Eigene	10	11	10	25	23	44	42	0	0	0
raspberry	5	43	33	200	198	213	129	6	5	1
smd	2	29	26	41	34	124	119	1	1	2
Gesamt	169	152	441	415	566	464	12	8	21	21
True Positiv	89.9%	94.1%	81.9%	66.7%	100%	40%	14.3%	16.7%	54.2%	



Problem: Farben werden verwechselt. Sehr dunkle Kondensatoren werden als Widerstände erkannt.

Große Anschlüsse werden mitunter mit großen geschirmten IC's verwechselt. Problem: Induktivitäten treten in zu vielen Formen und geringer Zahl auf, sodass diese nicht zuverlässig erkannt werden können.

Abschließend lassen sich viele der Probleme der Komponentenerkennung durch unzureichende Daten und Diversität dieser zusammenfassen. Dies gilt zum Beispiel für die geringe Zahl an Spule, Dioden, LEDs und Oszillatoren im Datensatz. Zudem müsste das Training um Bilder verschiedener Skalierung und Rotation erweitert werden, um gedrehte, sehr große oder sehr kleine Komponenten besser zu erkennen. Das letzte Problem bilden Ferritperlen (FB). Diese sind in kleiner Bauform fast unmöglich von schwarzen SMD-Widerständen zu unterscheiden.

Referenzen: „PCBSegClassNet“ von Dhruv Makwana, Sai Chandra Teja R. und Sparsh Mittal veröffentlicht im Journal „Expert Systems with Applications“ Volume 225; DOI: doi.org/10.1016/j.eswa.2023.120029
„FPIC: A Novel Semantic Dataset for Optical PCB Assurance“ von Neithan Jessurun, Olivia P. Olson-Paradis und anderen veröffentlicht im Journal „ACM Journal on Emerging Technologies in Computing Systems“ Volume 19; DOI: doi.org/10.1145/3588032
„Automated PCB Reverse Engineering“ von Stephan Kleber, Henrik Ferdinand Nölcher und Frank Kargl veröffentlicht auf der „WOOT'17: Proceedings of the 11th USENIX Conference on Offensive Technologies“ im August 2017
„FICS-PCB: A Multi-Modal Image Dataset for Automated Printed Circuit Board Visual Inspection“ von Hangwei Lu, Olivia Paradis und anderen im Juli 2020
„EE 368: Reverse Engineering of Printed Circuit Boards“ von Ben Johnson an der Stanford University im Jahr 2013

Wir danken Herrn Prof. Dr. Sven Tschirley und unserem Kommilitonen Adrian Strehlau für die freundliche Spende von bestückten und unbestückten Platinen zum Testen der Traceerkennung und Neuronalen Netze.