Semantische Segmentierung von Ankerkomponenten von Elektromotoren

Norbert Mitschke und Michael Heizmann

Karlsruher Institut für Technologie,
Institut für Industrielle Informationstechnik,
Hertzstraße 16, 76187 Karlsruhe

Zusammenfassung In diesem Beitrag wird die semantische Segmentierung von Ankern aus Elektromotoren und seinen Komponenten untersucht. Hierfür wird ein U-Net mit einem eigenständig angefertigten Datensatz trainiert, welcher aus Bildern von Ankern unterschiedlichster Bauformen besteht und im Rahmen dieses Beitrags angefertigt wurde. Aufgrund der geringen Anzahl von 75 Trainingsbildern werden neben einer geeigneten Standardaugmentierung auch eine neuartige Hintergrundaugmentierung und das Einbinden von Kanteninformationen untersucht. Mithilfe dieser Methoden kann der Testfehler bei der Segmentierung um insgesamt 70% reduziert werden.

Keywords Neuronale Netze, maschinelles Lernen, semantische Segmentierung, automatische Sichtprüfung

1 Einleitung

In diesem Beitrag wird ein Ansatz für die semantische Segmentierung der Komponenten von Altprodukten am Beispiel des Ankers von Elektromotoren vorgestellt. Die Segmentierung der Komponenten stellt den ersten Schritt für die Rückgewinnung von Altprodukten, dem sog. *Remanufcturing*, dar. Hierfür ist es erforderlich, die funktionsrelevanten Komponenten des Altproduktes zu erkennen, um diese anschließend inspizieren zu können. An die Erkennung ist eine hohe Anforderung an die Genauigkeit gebunden, da in weiteren Arbeiten auf Basis des Segmentierungsergebnisses nicht nur

die Lageparameter geschätzt werden, sondern das Ergebnis auch als Maske für das Zusammensetzen der Mantelfläche (engl. *stitching*) verwendet wird.

Somit ist ein möglichst robuster Klassifikator auf Pixelebene erforderlich, der einerseits Anker mit ungewissen Produktzuständen, die beispielsweise durch Defekte gegeben sind, und andererseits verschiedenste Ankerbauformen erkennt. In der Vergangenheit haben sich neuronale Netze [1] als vorteilhaft für komplexe Bildverarbeitungsaufgaben wie Klassifikation, Detektion oder Segmentierung herausgestellt. Speziell für die semantische Segmentierung von Bildern ist die Verwendung eines U-Net [2] der Stand der Technik.

Zunächst wird in Abschnitt 2 der Datensatz präsentiert, der für diesen Beitrag erstellt wurde. Anschließend wird in Abschnitt 3 der verwendete Ansatz vorgestellt. Dieser umfasst die Augmentierung in Abschnitt 3.1 und die Erweiterungen des U-Net in Abschnitt 3.3. Anhand des beschriebenen Versuchsaufbaus in Abschnitt 4 werden in Abschnitt 5 die Ergebnisse beschrieben. Der Beitrag schließt mit einer Zusammenfassung in Abschnitt 6.

2 Datensatz

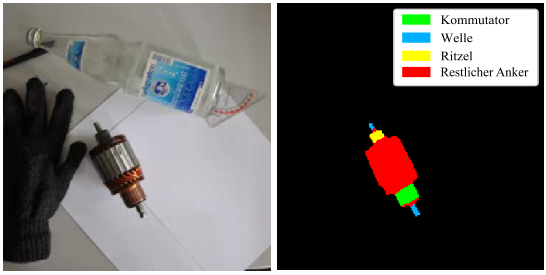
Das Lernen eines neuronalen Netzes erfordert eine Vielzahl an annotierten Bildern mit geeignetem Kontext. Für die Zwecke der Segmentierung von Ankern in Elektromotoren ist bisher kein öffentlich zugänglicher Datensatz verfügbar, weswegen ein relativ kleiner Datensatz mit insg. 96 Bildern erstellt wurde, da das Annotieren mit einem hohen Zeit- und Kostenaufwand verbunden ist. Der Datensatz wird im Folgenden beschrieben.

Der Datensatz besteht einerseits aus selbst aufgenommenen Bildern der am Institut vorliegenden Anker und andererseits aus frei zugänglichen Bildern aus dem Internet. Die eigenen Aufnahmen haben verschiedene irrelevante Objekte im Hintergrund, während die Bilder aus dem Internet oft von Online-Händlern stammen und einen einfarbigen Hintergrund aufweisen. Um die Bilder als Eingang für das neuronale Netz verwenden zu können, werden diese zu einem Quadrat beschnitten und anschließend mit einem geeigneten Aliasing-Filter auf 224×224 Pixel herunter- bzw. heraufgetastet. Von

den 96 Bildern werden 75 als Lernbilder und 21 als Testbilder verwendet.

Für das *Remanufcturing* sind drei Ankerkomponenten relevant. Diese sind der Kommutator (K), die Welle (W) und das Ritzel (R), deren Leitfähigkeit bzw. mechanische Eigenschaften starken Einfluss auf die Funktionsfähigkeit des Motors haben. Bei der Annotierung erhält jedes Pixel die Information, ob es zum Anker gehört und ggf. zu welcher Klasse es gehört (s. Abb. 2.1). Es ergibt sich somit ein Vier-Klassen-Problem innerhalb der Ankermaske, das die drei relevanten Klassen und eine Dummy-Klasse (X) enthält. Letztere beschriebt den restlichen Anker, d. T. Teile des Ankers, die keiner oben genannten Klasse zuzuordnen sind. Für die Menge aller Pixel des Ankers A und der Klassen K, W, R und X gilt

$$
\begin{aligned}
&K, W, R, X \in A,\\
&P \cap Q = \begin{cases} \emptyset, & P \neq Q,\\ P, & P = Q \end{cases}\\
&\qquad\qquad \text{für } (P, Q) \in (\{K, W, R, X\} \times \{K, W, R, X\}),\\
&K \cup W \cup R \cup X = A. \qquad (2.1)
\end{aligned}
$$



(a) Originalbild (b) Annotierung

Abbildung 2.1: Bei der Annotierung wird im Originalbild nach dem Anker und seinen Komponenten (Kommutator, Welle und Ritzel) gesucht.

3 Ansatz

In diesem Beitrag werden zwei Ansatzpunkte für die Verbesserung der Robustheit eines Detektors untersucht. Zum einen wird die Variationen der Anker durch Augmentierung beim Training erhöht. Dies kann als integrative Methode zur Erzeugung invarianter Merkmale verstanden werden [3]. Zum anderen wird Vorwissen verwendet, um triviale Fehler beim Lernen zu vermeiden und so das Training zu beschleunigen. Beide Ansätze werden im Folgenden für den in Abschnitt 2 beschriebenen Datensatz untersucht.

3.1 Augmentierung

Da die Annotierung von Bilddaten oft sehr zeit- oder kostenintensiv ist, sind die Lerndatensätze oft sehr klein, was in der Lernphase zu Überanpassung führt. Neben Regularisierung, Dropout und Batch-Normalisierung wird auch Bildaugmentierung verwendet. Hierbei wird eine Transformation auf ein Bild ausgeführt, die die Bildelemente manipuliert, während die Annotierung nur kohärent beeinflusst wird.

Für eine Segmentierung kommen fünf Arten von Augmentierung infrage: Spiegelung, affine Transformationen, Farbmanipulationen, Rauschen und Cutout. Hierbei müssen die Operationen an die Ankerbilder angepasst werden und können teilweise erweitert werden.

Eine zentrale Rolle spielt die Skalierung bei der affinen Transformation, da sie die Größe des Objektes bestimmt. Da die Bilder des Datensatzes Anker unterschiedlichster Größe enthalten, muss die Skalierung abhängig vom Bild so gewählt werden, dass die resultierende Größe des Ankers im Bild innerhalb einer gewissen Schwankungsbreite liegt. In diesem Betrag wird als Schwankungsbreite 10% bis 40% der Bildgröße gewählt, was ca. 5.000 bis 20.000 Pixeln entspricht.

Für alle Augmentierungsoperationen werden die Parameter stochastisch in geeigneten Grenzen gewählt. In den Experimenten wird eine sechsstufige Augmentierungspipeline verwendet, die aus den folgenden Stufen besteht:

- Spiegelung (keine, x-Achse, y-Achse, x- und y-Achse)

- Rotation ($-\frac{\pi}{2}$ bis $+\frac{\pi}{2}$)
- Skalierung durch Ausschneiden oder Padding (s. oben)
- Translation und Scherung entlang x- und y-Achse
- Cutout nach [4]
- Gaußfilter, Schärfung, Rauschen, Änderungen von Helligkeit bzw. Sättigung, Farbwertquantisierung oder Farbverschiebung

Da in den Bilder des Lerndatensatzes meist ein Anker als einziges Objekt vorhanden ist, besteht die Gefahr, dass das U-Net nur das Vorhandensein eines bloßen Objektes erlernt und es somit nicht vom spezifische Objekt, dem Anker, unterscheiden kann. Um dies zu vermeiden, wird Hintergrundaugmentierung verwendet. Hierzu wird die Grundwahrheit als binäre Maske $\mathbf{m_I}$ zur Extraktion des Ankers aus dem Bild $\mathbf{I}$ benutzt. Anschließend wird der extrahierte Anker vor einen zufälligen Hintergrund $\mathbf{H}_i$ aus dem dtd-Datensatz [5] gelegt. Das Ergebnis wird anschließend mit dem Tiefpassfilter $\mathbf{g_{Gauß}}$ gefiltert. Es ergibt sich

$$\mathbf{I_{aug}} = \mathbf{g_{Gauß}} * *(\mathbf{m_I} \odot \mathbf{I} + (1 - \mathbf{m_I}) \odot \mathbf{H}_i). \tag{3.1}$$

3.2 U-Net

Das U-Net nach [2] ist in Abb. 3.1 illustriert. Die Eingabe ist ein RGB-Bild und die Ausgabe gibt die geschätzte Klassenzugehörigkeit für jedes Pixel an. Die namensgebende Form des neuronalen Netzes entsteht durch die kleiner, aber dafür tiefer werdenden Merkmalskarten zur Mitte hin und die Querverbindungen, bei denen Merkmalskarten gleicher Größe konkateniert werden (gelbe Pfeile in Abb. 3.1).

Das verwendete U-Net hat eine Eingabegröße von 224×224 Pixeln, fünf Tiefenstufen und ca. 31 Mio. trainierbare Parameter. Auf jede 3×3-Faltungsschicht folgt Batch-Normalisierung nach [6] und eine ReLU-Aktivierung. Beim Hochtasten wird das 2×2-Interpolationsfilter auch im Training gelernt.

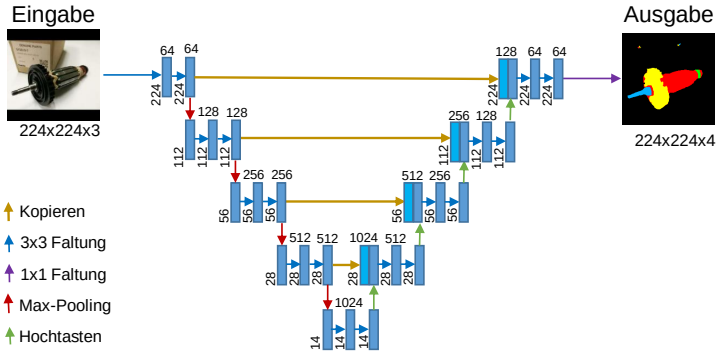


Abbildung 3.1: Die Abbildung zeigt den hier verwendeten Aufbau des U-Nets. Die oberen Zahlen an den blauen Rechtecken geben Anzahl der Merkmale bzw. die Tiefe der Aktivierungskarten an, während die seitlichen Zahlen die Höhe bzw. die Breite der Aktivierungskarte wiedergeben.

Als Zielfunktion wird der generalisierte Sörensen-Dice-Koeffizient c_{gSDK} nach [7] verwendet. Dieser bildet die gewichtete Summe der Sörensen-Dice-Koeffizienten oder einzelnen Klassen. Es gilt:

$$c_{\text{gSDK}} = \sum_i w_i \cdot c_i = \sum_i w_i \cdot \left(1 - \frac{|\mathbf{y} \cap \hat{\mathbf{y}}|}{|\mathbf{y} \cup \hat{\mathbf{y}}|}\right). \tag{3.2}$$

Mit dem Sörensen-Dice-Koeffizienten wird das negative Verhältnis von Schnitt zur Vereinigung zweier Flächen abgebildet. Nähert sich der Sörensen-Dice-Koeffizient dem Wert 0, so sind der Schnitt und die Vereinigung identisch.

3.3 Erweiterung des U-Net

Neben Augmentierung eignen sich zusätzliche Informationen, um die Genauigkeit des neuronalen Netzes zu erhöhen. In diesem Abschnitt werden Methoden aufgeführt, die das U-Net um Zusatzinformation erweitern.

Die Hinzunahme der Kanteninformation in einer der hinteren Schichten kann zu einer Verbesserung führen, da die Objektgrenzen des Ankers im Bild mit den Kanten im Bild zusammenfallen.

Zur Kantenextraktion wird der Marr-Hildreth-Operator verwendet. Das Ergebnis wird anschließend normiert. Die Kante wird nach der obersten Konkatenierungsschicht entweder hinzuaddiert oder angehängt. Die schematische Veränderung des U-Net ist in Abb. 3.2 dargestellt.

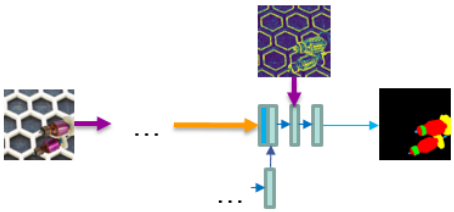


Abbildung 3.2: Die Abbildung zeigt, wie die letzten Schichten des U-Net abgeändert werden, um die Kanteninformationen einzubringen.

Da es sich bei den Ankern um zusammenhängende Objekte handelt, ist eine Regularisierung sinnvoll, die lange Konturen bestraft. Somit können Löcher, kleine Fehldetektionen oder andere Artefakte reduziert werden. Hierfür eignet sich die *Total-Variation*-Regularisierung (TV-Regularisierung) nach [8]. Der Strafterm L_{TV} für das Segmentierungsergebnis $\mathbf{A} = [\mathbf{a}^{(\mathrm{A})}, \mathbf{a}^{(\mathrm{K})}, \mathbf{a}^{(\mathrm{W})}, \mathbf{a}^{(\mathrm{R})}] \in \mathbb{R}^{(224,224,4)}$, wobei das x in $a^{(\mathrm{x})}$ für die Aktivierungskarte des Ankers (A), des Kommutators (K), der Welle (W) oder des Ritzels (R) steht, wird gewählt zu

$$L_{\mathrm{TV}} = \lambda_{\mathrm{TV}} \sum_{n=\{\mathrm{A,K,W,R}\}} w_n \cdot \sum_i \sum_j \mathbf{a}^{(n)}[i,j] - \frac{1}{2}\mathbf{a}^{(n)}[i+1,j] - \frac{1}{2}\mathbf{a}^{(n)}[i,j+1]. \tag{3.3}$$

Eine Erweiterung hiervon ist, den Strafterm an Stellen, an denen eine Kante vorliegt, zu verkleinern.

Im originalen U-Net wird das Ergebnis nach der letzten Faltungsschicht mit einer Sigmoid-Funktion $\sigma(\cdot)$ aktiviert. Dies kann potentiell dazu führen, dass sich die Klassen nicht gegenseitig ausschließen oder einzelne Teile einer Komponente wie bspw. der Welle zwar als

Welle erkannt werden, aber nicht als Teil des Ankers erkannt werden. Dies kann durch zusätzliche Restriktionen vermieden werden, die allerdings die Konvergenzeigenschaften des Netzes beeinflussen. Im Folgenden werden zwei Alternativen vorgestellt, das Ergebnis der letzten Faltungsschicht $\mathbf{A}$ zu aktivieren.

Der erste Ansatz ist ein multiplikativer Ansatz mit Sigmoid-Aktivierung (MSig), der ausschließt, dass Komponenten außerhalb der Ankerklasse liegen. Die Aktivierungsvorschrift für die Klasse x lautet

$$\begin{aligned} &\text{für } x = A\text{: } \mathbf{b}^{(\mathrm{A})} = \sigma(\mathbf{a}^{(\mathrm{A})}) \\ &\text{für } x \neq A\text{: } \mathbf{b}^{(x)} = \mathbf{b}^{(\mathrm{A})} \cdot \sigma(\mathbf{a}^{(x)}). \end{aligned} \tag{3.4}$$

Beim zweiten Ansatz (MSmax) wird das gegenseitige Ausschließen der Klassen durch eine Softmax-Funktion S sichergestellt. Es ergibt sich

$$\begin{aligned} &\text{für } x = A\text{: } \mathbf{b}^{(\mathrm{A})} = \sigma(\mathbf{a}^{(\mathrm{A})}) \\ &\text{für } x \neq A\text{: } \mathbf{b}^{(x)} = \mathbf{b}^{(\mathrm{A})} \cdot S([\mathbf{a}^{(x)}, \mathbf{0}]). \end{aligned} \tag{3.5}$$

Für den Fall eines Pixels im Anker ohne Zugehörigkeit zur Klasse K, W oder R wird für MSmax eine Aktivierungskarte mit dem konstanten Wert 0 hinzugefügt. Dies entspricht der Klasse X in Abschnitt 2.

4 Versuchsaufbau

Jedes Einzelexperiment wird dreimal wiederholt. Das Ergebnis wird gemittelt. Bei jedem Durchlauf wird das U-Net für 100 Durchläufe zu je 16×1024 augmentierten Bildern mit dem *Nadam*-Optimierer trainiert. Es wird ein kosinusartiger Rückgang der Lernrate mit einer Anfangslernrate von 10^{-3} verwendet. Als Vergleichsmetrik wird der generalisierte Sörensen-Dice-Koeffizient und der Jaccard-Koeffizient der einzelnen Komponenten verwendet.

Zuerst wird der Einfluss von Augmentierung untersucht. Es werden vier Stufen der Augmentierung mit und ohne Hintergrundaugmentierung untersucht. Bei der Hintergrundaugmentierung wird

mit der Wahrscheinlichkeit 0,6 ein zufälliger Hintergrund verwendet, ansonsten bleibt der Originalhintergrund bestehen. In der untersten Stufe (Stufe 0) wird keine Augmentierung durchgeführt. In Stufe I wird die Basisaugmentierungskaskade verwendet, die aus Spiegelung, Rotation, Skalierung, Translation, Scherung und Cutout besteht. Für Stufe II wird diese Kaskade gemäß Abschnitt 3.1 um eine Stufe mit den Operationen des letzten Stichpunktes von Abschnitt 3.1 erweitert. In der letzten Stufe (Stufe III) wird die Cutout-Stufe um zwei eigene Verfahren erweitert. Zum einen werden zufällig einzelne Komponenten verdunkelt und zum anderen wird Cutout mehrfach mit kleineren Rechtecken angewendet.

Danach werden mit der besten Augmentierungsstrategie die Verfahren aus Abschnitt 3.3 verglichen. Zunächst wird der Einfluss der Aktivierung der letzten Schicht untersucht. Anschließend werden verschiedene Kombinationen aus TV-Regularisierung und Hinzufügen von Kanteninformationen betrachtet.

5 Ergebnisse

Im Folgenden werden die Ergebnisse für die Augmentierung und für Erweiterungen des U-Net vorgestellt.

5.1 Augmentierung

Die Ergebnisse sind in Tabelle 1 dargestellt und zeigen, dass sich der Sörensen-Dice-Koeffizient bei der Verwendung eines zufälligen Hintergrunds bei allen Augmentierungsstufen um ca. 40% verbessert. Dies stützt die These, dass durch einen zufälligen Hintergrund der Fokus des Trainings auf das relevante Objekte verlagert wird, woraus eine bessere Generalisierung des Netzes folgt. Ohne Augmentierung des Objekts führt Hintergrundaugmentierung zu einer Verschlechterung von 36%, da die Position des Objektes vom Netz auswendig gelernt werden kann.

Bei der hier getroffenen Auswahl der Augmentierungskaskade führt eine stärkere Augmentierung zu leicht besseren Sörensen-Dice-Koeffizienten. Daher wird das beste Segmentierungsergebnis bei Stufe III mit Hintergrundaugmentierung erzielt.

Tabelle 1: Ergebnisse der Augmentierung. Es ist der generalisierte Sörensen-Dice-Koeffizient des Gesamtergebnisses sowie der Jaccard-Koeffizient der Komponenten angeben.

	III	II	I	0	III	II	I	0
	Originaler Hintergrund				Zufälliger Hintergrund			
gSDK	0,179	0,178	0,188	0,328	**0,100**	0,102	0,103	0,447
Anker	0,885	0,887	0,886	0,762	**0,952**	0,951	**0,952**	0,639
Kommutator	0,619	0,616	0,595	0,389	0,650	**0,656**	0,657	0,368
Welle	0,457	0,452	0,451	0,271	**0,560**	0,552	0,548	0,243
Ritzel	0,359	0,357	0,316	0,247	0,461	**0,462**	0,455	0,179

5.2 Erweiterung des U-Nets

Für die Standardaktivierung ergibt sich ein gSDK von 0,100. Die beiden anderen Aktivierungen liefern ein gSDK von ebenfalls 0,100 (MSig) bzw. von 0,105 (MSmax). Trotz der Beseitigung aller logischen Widersprüche verschlechtert sich das Ergebnis. Für die weitere Analyse werden daher nur die Sigmoid-Aktivierung und MSig miteinander verglichen, auch weil für bestimmte Kombinationen von MSmax der Jaccard-Koeffizient der Welle nicht konvergiert. Insgesamt zeigt sich, dass die logischen Zusammenhänge beim Training eigenständig erlernt werden.

Tabelle 2: Ergebnisse bei Verwendung der Zusatzinformationen. Mit *K* ist die Konkatenierung und mit *A* die Addition der Kanten gemeint. *R* bedeutet reguläre TV-Regularisierung und *G* die mit Kanten gewichtete. Bei – wird keine Kanteninformation bzw. TV-Regularisierung verwendet.

	Sigmoid-Aktivierung								
Kante	-	-	-	K	K	K	A	A	A
TV-Reg.	-	R	G	-	R	G	-	R	G
gSDK	0,100	0,101	0,104	**0,096**	0,102	0,104	0,100	0,102	0,103
	MSig								
Kante	-	-	-	K	K	K	A	A	A
TV-Reg.	-	R	G	-	R	G	-	R	G
gSDK	0,100	0,101	0,100	0,099	0,102	0,100	0,097	0,102	0,099

Die Ergebnisse sind in Tab. 2 zusammengefasst. Insgesamt sind die erzielten Verbesserungen mit bis zu 5% eher moderat. Eine TV-

Regularisierung hat eher negative Auswirkungen auf das Ergebnis, während Kanteninformationen neutrale bis positive Auswirkungen haben. Am besten schneidet das Verfahren mit Sigmoid-Aktivierung und Kantenkonkatenierung ab.

6 Zusammenfassung

In diesem Beitrag wird ein Segmentierungsnetz für Anker von Elektromotoren vorgestellt, bei dem relevante Komponenten vom Rest des Ankers und dem Hintergrund getrennt werden, um diese anschließend inspizieren zu können. Durch Augmentierung und insbesondere der hier vorgestellten Hintergrundaugmentierung kann das Ergebnis signifikant verbessert werden. Mithilfe von Kanteninformationen kann die Genauigkeit um weitere 4% erhöht werden.

Mit den erzielten Ergebnissen können im Anschluss Lageparameter wie Rotation oder perspektivische Verzerrung des Ankers geschätzt werden. Dies ermöglicht eine bildbasierte Regelung für die optimale Ausrichtung einer positionierbaren Kamera und eine Extraktion der Mantelfläche der relevanten Komponenten.

In weiteren Arbeiten soll das U-Net deutlich länger mit den gefundenen Parametern angelernt und anschließend für die Segmentierung von Videos bzw. Echtzeit-Kamerasystemen verwendet werdet. Mithilfe eines internen Modells soll das Segmentierungsergebnis stabilisiert werden und die Größe des Ankers im jeweiligen Eingabebild durch *Zero-Padding* oder Heranzoomen auf die im Training festgelegte Größe geregelt werden.

Danksagung

Das Projekt AgiProbot wird durch die Carl-Zeiss-Stiftung gefördert.

Literatur

1. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

2. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional Networks for Biomedical Image Segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.
3. H. Schulz-Mirbach, "Constructing Invariant Features by Averaging Techniques," in *12th IAPR International Conference on Pattern Recognition, Conference B: Patern Recognition and Neural Networks, ICPR 1994, Jerusalem, Israel, 9-13 October, 1994, Volume 2*, 1994, pp. 387–390. [Online]. Available: https://doi.org/10.1109/ICPR.1994.576950
4. T. Devries and G. W. Taylor, "Improved Regularization of Convolutional Neural Networks with Cutout," *CoRR*, vol. abs/1708.04552, 2017. [Online]. Available: http://arxiv.org/abs/1708.04552
5. M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing Textures in the Wild," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.
6. S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.
7. R. Crum, O. Camara, and D. Hill, "Generalized Overlap Measures for Evaluation and Validation in Medical Image Analysis," *IEEE Transactions on Medical Imaging*, vol. 25, no. 11, pp. 1451–1461, 2006.
8. D. Strong and T. Chan, "Edge-Preserving and Scale-Dependent Properties of Total Variation Regularization," *Inverse Problems*, vol. 19, no. 6, p. S165, 2003.