



Metody klasyfikacji danych zaszumionych

Stanisław Kaźmierczak

Agenda

- Szum
- Źródła szumu
- Zaszumianie i odszumianie
- Nauka i testowanie
- Architektura sieci
- Wyniki
- Wnioski oraz kierunki dalszych badań



Szum

- Definicja
 - *Cokolwiek co zakłóca, przyciemnia relację między cechami obiektu, a jej klasą [1, 3]*
 - *Niesystematyczny błąd [2, 3]*
- Wpływa negatywnie na działanie klasyfikatorów
- Szum jest wszechobecny i stanowi wyzwanie w wielu praktycznych zastosowaniach
 - Przykład: badania medyczne, których jakość i dokładność jest bardzo różna
- Potrzebne są techniki, które eliminują szum bądź redukują jego konsekwencje
 - Jest to tym bardziej uzasadnione, gdyż wysokiej jakości otykietowane dane są drogie, a ich uzyskanie - czasochłonne



Szum

- Rodzaje szumu [2, 4]
 - Szum związany z atrybutami (ang. *Feature/attribute noise*)
 - Szum związany z etykietami (ang. *Label/class noise*)
- Szum na etykietach jest potencjalnie bardziej szkodliwy [4]
 - Jest wiele cech, ale tylko jedna etykieta
 - Znaczenie (istotność) każdej cechy na proces uczenia jest różna, natomiast etykiety mają zawsze duży wpływ
- Potrzebne są techniki, które eliminują szum bądź redukują jego konsekwencje
 - Jest to tym bardziej uzasadnione, gdyż wysokiej jakości otykietowane dane są drogie, a ich uzyskanie - czasochłonne



Źródła szumu na etykietach

- Informacja dostarczona ekspertowi jest niewystarczająca do przeprowadzenia niezawodnego etykietowania
 - Przykład: decyzja o chorobie na podstawie wywiadu lekarskiego
- Błąd eksperta (człowieka lub maszyny)
 - Etykiety dostarczone przez nie-ekspertów, np. *Amazon Mechanical Turk*
- Nieobiektywność ekspertów
 - Przykład: eksperci rzadko zgadzają się odnośnie dokładnych granic we wzorcach sygnału w analizie EKG
- Sposób kodowania lub problemy komunikacyjne
 - Przykład: błędne oznaczenie maili jako spam wynikające z nierozumienia mechanizmu lub przypadkowego kliknięcia
 - Szacuje się, że ok. 5% danych przechowywanych w bazach danych posiada błędy związane z kodowaniem [5]



Źródła szumu na atrybutach

- Niedostateczna jakość urządzeń pomiarowych
- Wynikające z praw fizyki
 - Przykład: wibrujące pole elektromagnetyczne
- Warunki pomiaru
 - Przykłady: hałas w tle, słabe oświetlenie
- Konsekwencja kompresji danych
- ...



T.S. Nazaré, G.B.P. da Costa, W.A. Contato, M. Ponti,

„Deep Convolutional Neural Networks and Noisy Images”

In: M. Mendoza, S. Velastín (eds), *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2017. Lecture Notes in Computer Science*, vol 10657. Springer, Cham

Zbiory danych

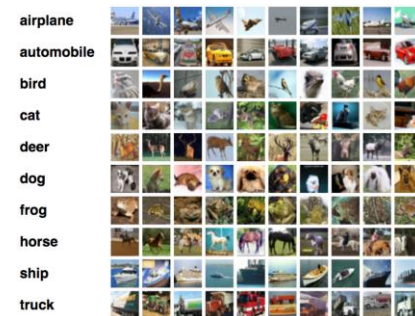
- **MNIST**

- Zbiór pisanych ręcznie cyfr, jeden z głównych benchmarków dla zadania klasyfikacji



- **CIFAR-10**

- Składa się z 60k kolorowych obrazków 32x32 równo podzielonych na 10 klas



- **SVHN**

- Zbiór prawie 100k obrazków z numerami domów zaczerpniętych z *Google Street View*



Zaszumianie

- 5 kopii oryginalnych zbiorów zostało zaszumianych przy pomocy szumu gaussowskiego z odchyleniem standardowym $\sigma = \{10, 20, 30, 40, 50\}$
- Kolejnych 5 kopii zostało zaszumionych przy pomocy szumu typu pieprz i sól z parametrem $p = \{0.1, 0.2, 0.3, 0.4, 0.5\}$ oznaczającym prawdopodobieństwo zmiany wartości piksela do wartości ekstremalnej



[7]

Odszumianie

- Szum typu sól i pieprz
 - Filtr medianowy
- Szum gaussowski
 - Algorytm średnich nielokalnych (*Non-local means* - *NLM*) [8]
- 21 różnych wersji każdego zbioru danych
 - Oryginalny, 10 zaszumionych, 10 odszumionych

Algorytm NLM

Ω – obraz wejściowy

$u(p)$ – nowa (po zastosowaniu filtra) wartość piksela p

$v(q)$ – wejściowa wartość piksela q

$f(p, q)$ – funkcja wazaca

$C(p)$ – czynnik normalizujący

$B(p)$ – lokalna średnia wartości pikseli otaczających piksel p

h – parametr filtrujący (np. odchylenie standardowe)

$R(p)$ – kwadratowy region pikseli otaczający piksel p

$$u(p) = \frac{1}{C(p)} \sum_{q \in \Omega} v(q) f(p, q)$$

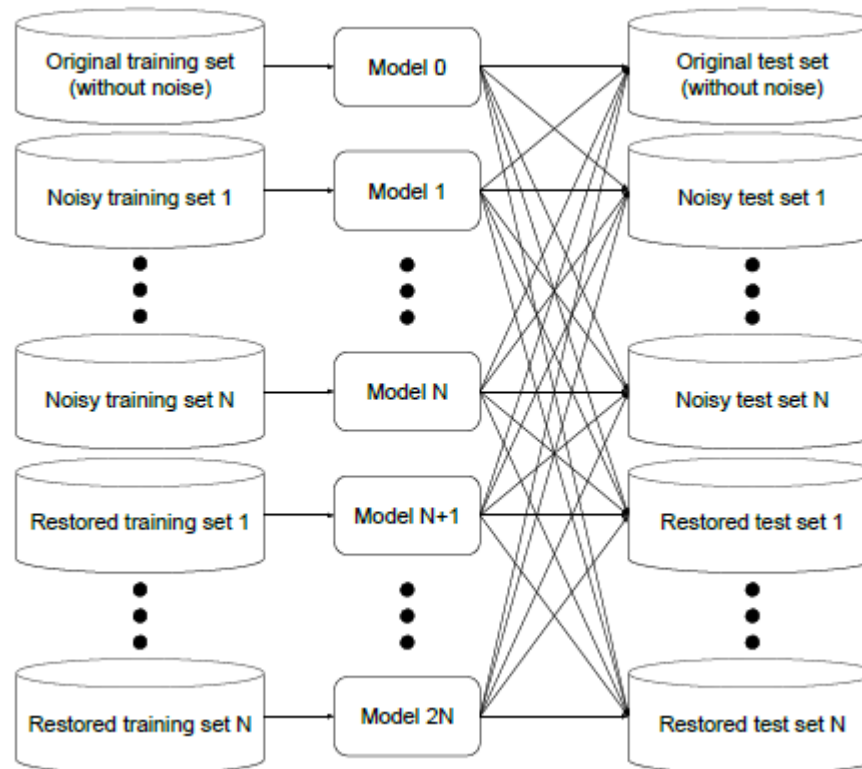
$$C(p) = \sum_{q \in \Omega} f(p, q)$$

$$f(p, q) = e^{-\frac{|B(q) - B(p)|^2}{h^2}}$$

$$B(p) = \frac{1}{|R(p)|} \sum_{i \in R(p)} v(i)$$

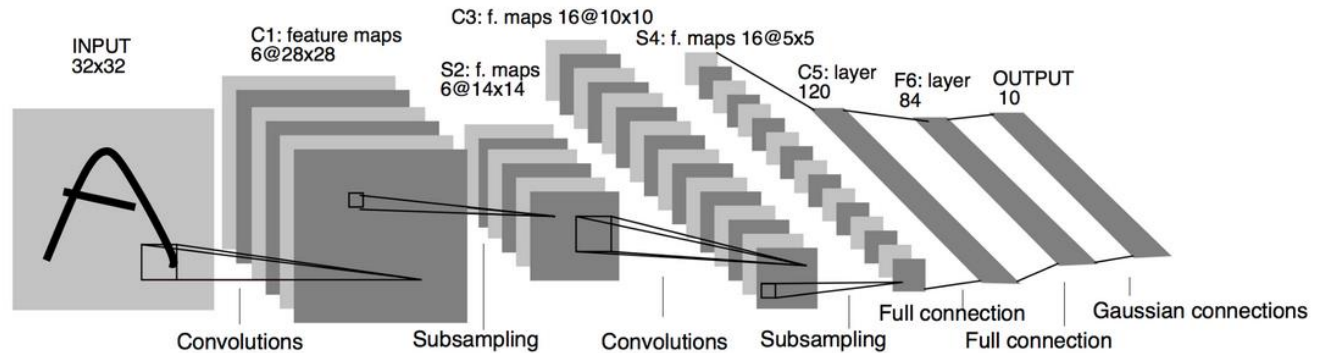
Nauka i testowanie

- Sieć była osobno trenowana dla każdej wersji danych, tworząc w ten sposób 21 klasyfikatorów
- Każdy klasyfikator testowany był na wszystkich wersjach zbioru testowego



Architektury sieci

- *MNIST* – architektura zbliżona do LeNet-5 [9]



- CIFAR-10 i SVHN – architektura zbliżona do modelu C z [10]

Model		
A	B	C
	Input 32 × 32 RGB image	
5 × 5 conv. 96 ReLU	5 × 5 conv. 96 ReLU 1 × 1 conv. 96 ReLU	3 × 3 conv. 96 ReLU 3 × 3 conv. 96 ReLU
	3 × 3 max-pooling stride 2	
5 × 5 conv. 192 ReLU	5 × 5 conv. 192 ReLU 1 × 1 conv. 192 ReLU	3 × 3 conv. 192 ReLU 3 × 3 conv. 192 ReLU
	3 × 3 max-pooling stride 2	
	3 × 3 conv. 192 ReLU	
	1 × 1 conv. 192 ReLU	
	1 × 1 conv. 10 ReLU	
	global averaging over 6 × 6 spatial dimensions	
	10 or 100-way softmax	

Peak signal-to-noise ratio (PSNR)

- Stosunek maksymalnej mocy sygnału do mocy szumu zakłócającego ten sygnał
- Ze względu na szeroki zakres wartości PSNR wyrażany jest w decybelach

$$MSE = \frac{1}{N \cdot M} \sum_{i=1}^N \sum_{j=1}^M ([f(i, j) - f'(i, j)]^2)$$

N, M – wymiary obrazu w pikselach

f(i, j) – wartość piksela o współrzędnych (i, j) oryginalnego obrazu

f'(i, j) – wartość piksela o współrzędnych (i, j) zaszumionego obrazu

max(f(i, j)) = wartość maksymalna danego sygnału

$$PSNR = 10 \cdot \log_{10} \frac{[\max(f(i, j))]^2}{MSE}$$

Peak signal-to-noise ration (PSNR)

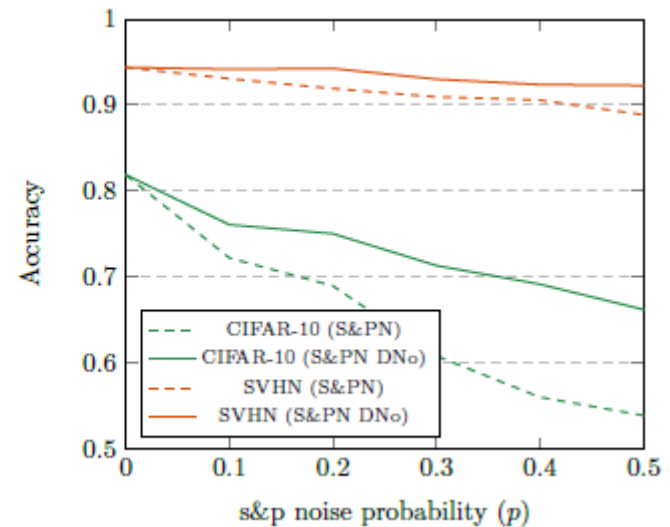
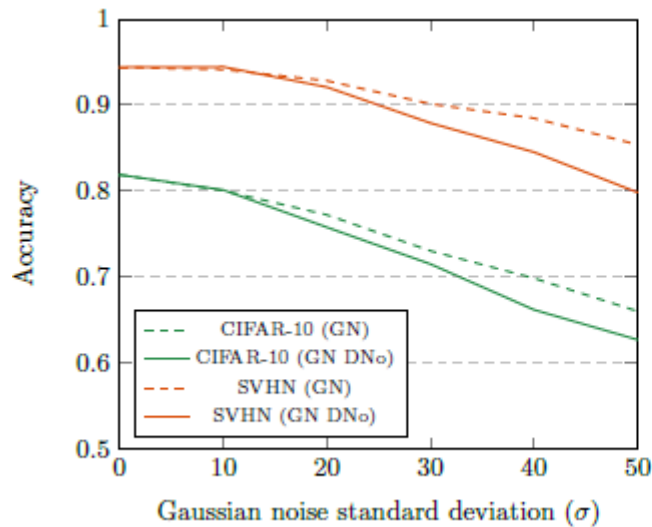
Noise		MNIST		CIFAR-10		SVHN	
		noisy	restored	noisy	restored	noisy	restored
Gaussian	$\sigma = 10$	30.93	33.18	28.25	29.88	28.15	32.65
	$\sigma = 20$	24.86	27.46	22.36	25.54	22.21	28.15
	$\sigma = 30$	21.35	24.35	18.99	22.81	18.81	25.21
	$\sigma = 40$	18.87	21.54	16.67	20.85	16.47	23.40
	$\sigma = 50$	16.96	17.84	14.95	19.49	14.74	22.26
s&p	$p = 0.1$	13.44	20.78	15.23	25.89	15.52	34.30
	$p = 0.2$	10.79	18.54	12.50	24.00	12.79	29.32
	$p = 0.3$	9.36	16.69	10.98	21.64	11.28	24.82
	$p = 0.4$	8.43	15.13	9.97	19.22	10.28	21.28
	$p = 0.5$	7.78	13.84	9.23	17.04	9.55	18.53

$$PSNR = 10 \cdot \log_{10} \frac{[\max(f(i, j))]^2}{MSE}$$

Wyniki klasyfikacji

Noise Type	MNIST		CIFAR-10		SVHN		
	noisy	restored	noisy	restored	noisy	restored	
original	0.9903		0.8192		0.9438		
Gaussian	$\sigma = 10$	0.9916	0.9885	0.8007	0.8016	0.9415	0.9445
	$\sigma = 20$	0.9875	0.9878	0.7727	0.7583	0.9284	0.9210
	$\sigma = 30$	0.9898	0.9867	0.7309	0.7156	0.9015	0.8793
	$\sigma = 40$	0.9890	0.9851	0.6991	0.6625	0.8849	0.8455
	$\sigma = 50$	0.9860	0.9814	0.6608	0.6277	0.8542	0.7987
s&p	$p = 0.1$	0.9799	0.9861	0.7227	0.7613	0.9308	0.9418
	$p = 0.2$	0.9793	0.9802	0.6902	0.7508	0.9193	0.9425
	$p = 0.3$	0.9753	0.9718	0.6088	0.7138	0.9095	0.9301
	$p = 0.4$	0.9641	0.9605	0.5610	0.6921	0.9057	0.9239
	$p = 0.5$	0.9437	0.9426	0.5398	0.6627	0.8889	0.9228

Dokładność (trening i testowanie na tej samej wersji zaszumienia)



* Różnice dla zbioru MNIST są bardzo małe

Wyniki klasyfikacji

- Pomimo wzrostu PSNR po zastosowaniu NLM w celu odszumiania, dokładność klasyfikacji spadła (względem wersji zaszumionej)
- Wynika to prawdopodobnie z faktu, że wspomniana procedura odszumiania generuje nieostre obrazki, tracąc w ten sposób istotne informacje

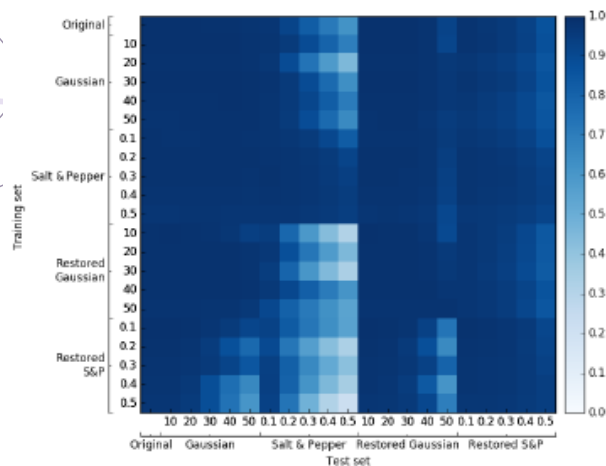


Rząd pierwszy: oryginalne obrazki

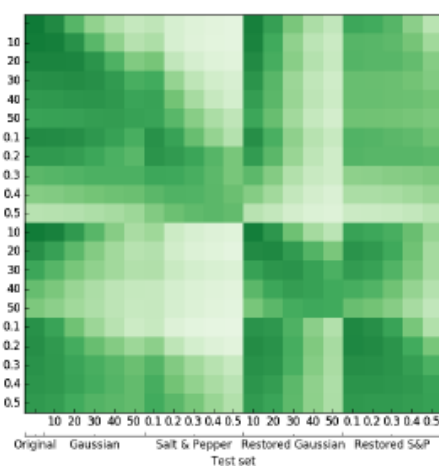
Rząd drugi: obrazki z szumem gaussowskim o odchyleniu standardowym 30 i odzyskane metodą NLM

Rząd trzeci: obrazki z szumem typu sól i pieprz ($p = 0.3$) i odzyskane przy użyciu filtra medianowego

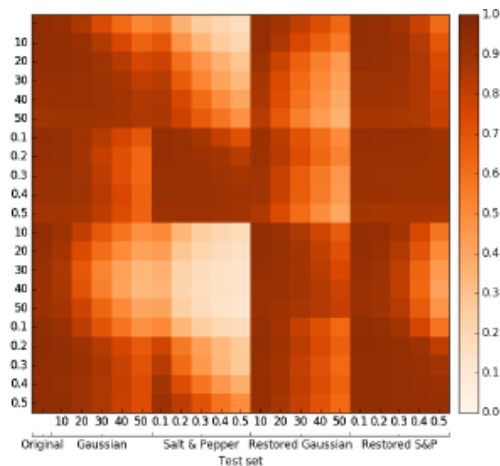
Wyniki klasyfikacji



MNIST



CIFAR-10



SVHN

- Najlepsza dokładność uzyskiwana jest, gdy klasyfikowane dane są tej samej jakości (ten sam stopień zaszumienia), co dane treningowe
- W zależności od zbioru treningowego osiągnięto różne zdolności generalizacji (im ciemniejsze wiersze tym lepiej)
 - Przykład: MNIST, szum typu sól i pieprz, bez odszumiania

Wyniki klasyfikacji

Noise Type	MNIST		CIFAR-10		SVHN		
	noisy	restored	noisy	restored	noisy	restored	
original	93.13 ± 9.68		43.50 ± 21.84		68.21 ± 24.64		
Gaussian	$\sigma = 10$	94.59 ± 7.35	87.68 ± 18.68	45.04 ± 21.80	48.83 ± 21.12	71.87 ± 22.47	67.41 ± 25.15
	$\sigma = 20$	91.00 ± 14.08	90.77 ± 14.18	49.52 ± 20.41	48.68 ± 21.06	73.72 ± 20.28	67.41 ± 25.15
	$\sigma = 30$	94.26 ± 8.65	88.29 ± 18.43	50.72 ± 18.57	47.75 ± 18.48	74.36 ± 18.94	61.09 ± 28.32
	$\sigma = 40$	94.49 ± 6.94	92.95 ± 9.90	51.76 ± 16.79	41.82 ± 18.07	75.62 ± 17.54	61.01 ± 28.67
	$\sigma = 50$	93.58 ± 8.16	90.72 ± 12.42	51.23 ± 14.85	38.31 ± 16.04	75.18 ± 16.76	63.51 ± 27.81
s&p	$p = 0.1$	95.85 ± 4.02	90.12 ± 12.55	56.78 ± 15.04	44.20 ± 22.47	82.97 ± 11.81	66.69 ± 24.95
	$p = 0.2$	97.13 ± 2.12	84.71 ± 19.06	56.92 ± 12.89	51.69 ± 20.08	83.41 ± 11.47	76.33 ± 20.13
	$p = 0.3$	97.27 ± 1.74	88.13 ± 15.61	49.79 ± 11.89	57.20 ± 14.97	82.26 ± 13.01	77.27 ± 19.79
	$p = 0.4$	97.11 ± 1.57	83.70 ± 18.57	42.83 ± 11.85	57.83 ± 13.71	82.84 ± 13.28	80.14 ± 16.04
	$p = 0.5$	96.32 ± 1.87	81.66 ± 22.40	33.21 ± 12.21	57.79 ± 11.76	80.02 ± 14.14	82.16 ± 12.89

Średnia (po wszystkich zbiorach testowych) dokładność i jej odchylenie standardowe dla każdego modelu (zbioru treningowego)

- Sieci uczone zaszumionymi obrazkami uzyskują lepsze rezultaty niż sieć uczona oryginalnymi obrazkami
 - MNIST: oryginalny – 93.13 +/- 9.68%, najlepszy – 97.27 +/- 1.74%
 - CIFAR: oryginalny – 43.50 +/- 21.84%, najlepszy – 57.83 +/- 13.71%
 - SVHN: oryginalny – 68.21 +/- 24.64%, najlepszy – 83.41 +/- 11.47%



Wnioski

- W sytuacjach, gdy mamy do czynienia z obrazami o zróżnicowanej jakości, trenowanie sieci danymi, które zostały sztucznie zaszumione, może przynieść pozytywne rezultaty
- Zwiększa to odporność sieci na inne typy i poziomy szumu
- Odszumianie metodą filtra medianowego obrazków z szumem typu sól i pieprz poprawia dokładność klasyfikacji danych tej samej jakości
- Modele trenowane obrazkami z szumem typu sól i pieprz z reguły uzyskiwały najlepszą średnią dokładność (najlepsza odporność na inne szumy)
- Odszumianie metodą NLM z reguły pogarszało rezultaty
- Przyczyną jest prawdopodobnie usunięcie z obrazków istotnych informacji będące konsekwencją wygładzającego działania filtra
- Lepsze rezultaty być może da się osiągnąć dobierając lepsze parametry filtra



Kierunek dalszy badań

- Zastosowanie głębszych modeli jak np. sieci rezydualne czy VGG
- Przeprowadzenie eksperymentów na większych zbiorach danych, np. ImageNet



Źródła

1. R. J. Hickey, “Noise modelling and evaluating learning from examples”, *Artif. Intell.*, vol. 82, no. 1-2, pp. 157–179, 1996.
2. J. R. Quinlan, “Induction of decision trees”, *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.
3. B. Frenay and M. Verleysen, “Classification in the Presence of Label Noise: A Survey” in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 845-869, May 2014.
4. X. Zhu and X. Wu, “Class noise vs. attribute noise: A quantitative study”, *Artif. Intell. Rev.*, vol. 22, pp. 177–210, 2004.
5. K. Orr, “Data quality and systems theory”, *Commun. ACM*, vol. 41, no. 2, pp. 66–71, 1998.
6. T.S. Nazaré, G.B.P. da Costa, W.A. Contato, M. Ponti, „Deep Convolutional Neural Networks and Noisy Images”, In: M. Mendoza, S. Velastín (eds), *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2017. Lecture Notes in Computer Science*, vol 10657. Springer, Cham

Źródła

7. <http://www.fit.vutbr.cz/~vasicek/imagedb/>
8. A. Buades, B. Coll and J. -. Morel, "A non-local algorithm for image denoising," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2005, pp. 60-65 vol. 2.
9. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE 86(11)*, 2278-2324, 1998.
10. J.T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, "Striving for simplicity: The all convolutional net", *arXiv preprint arXiv:1412.6806*, 2014.