

AKSELERASI *MAXPOOL* PADA PENGENALAN OBYEK DAN PENGUKURAN PANJANG *KATSUWONUS PELAMIS, EUTHYNNUS AFFINIS, CORYPHAENA HIPPURUS* DAN *LOLIGO CHINENSIS* MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK*

(*MAXPOOL ACCELERATION FOR OBJECT RECOGNITION AND LENGTH MEASUREMENT OF KATSUWONUS PELAMIS, EUTHYNNUS AFFINIS, CORYPHAENA HIPPURUS AND LOLIGO CHINENSIS USING CONVOLUTIONAL NEURAL NETWORK*)

Ujian Terbuka

Suryadiputra Liawatimena (2040002631)

Promotor: Prof. Dr. Edi Abdurahman, MS., MSc

Co-Promotor 1: Agung Trisetyarso, S.Si., M.Si., Ph.D

Co-Promotor 2: Dr. Eng. Antoni Wibowo, S.Si., M.Kom., M.Eng

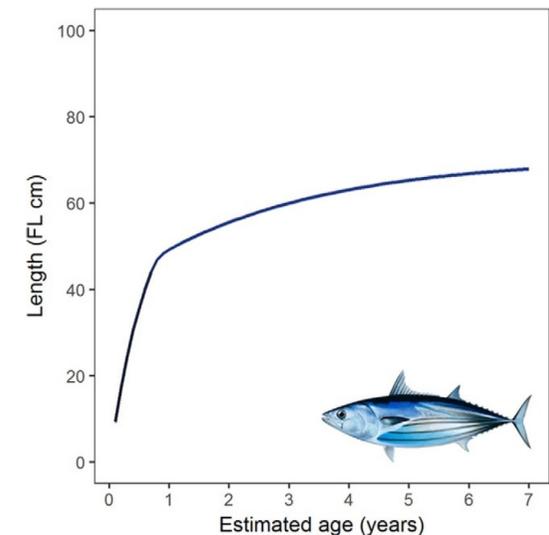
Doctor of Computer Science Program
Binus Graduate Program
Jakarta 2021

Latar Belakang

- Ikan Tuna, Cakalang dan Tongkol (TCT) merupakan komoditas dengan nilai ekspor terbesar kedua dari hasil perikanan Indonesia (~714 juta USD di 2018). Namun yang penting menjadi perhatian bersama adalah karena berdasarkan data yang dikeluarkan *Indian Ocean Tuna Commission* (IOTC), penangkapan tuna di hampir seluruh wilayah perairan Indonesia sudah dinyatakan **melebihi batas overfishing** dan itu sudah berjalan dari 2010 hingga 2014. Dua dari tiga jenis tuna yang ada (Madidihang dan Cakalang) dinyatakan bisa punah dalam waktu tiga hingga 10 tahun jika tidak segera dilakukan pembatasan penangkapan (Ambari, 2017).
- Salah satu tanda terjadinya *overfishing* adalah ketika ikan hasil tangkapan belum mencapai usia produktif yang dapat diketahui dari panjang ikan tersebut.

Latar Belakang (lanjutan)

- Menurut Artetxe-Arrate et al (2021) panjang ikan berdasarkan umur ikan Cakalang dapat dilihat pada Gambar 2.19. Model pertumbuhan Ikan Cakalang dicirikan pertumbuhan yang cepat pada tahap pertama, diikuti oleh pertumbuhan yang lebih lambat dalam tahap kedua. Karena tinggi awal tingkat pertumbuhan, cakalang dapat mencapai 45cm FL pada tahun pertama kehidupan, dan antara 50 dan 65cm FL di tahun kedua, dari mana tingkat pertumbuhan melambat. Panjang ikan Cakalang dengan umur tua berkisar hingga 70 cm.
- Ikan Cakalang dianggap memasuki umur produktif setelah mencapai 2 tahun dengan angka harapan hidupnya sampai 7 tahun. Panjang Ikan Cakalang usia produktif 50-70 cm. Mengenai berat ikan tergantung dari lingkungan hidupnya sehingga dengan panjang yang sama tapi memiliki berat yang berbeda.



Latar Belakang (lanjutan)

- Salah satu tugas dan fungsi Kementerian Kelautan dan Perikanan (KKP) adalah pelaporan statistik kelautan dan perikanan (Undang-Undang Republik Indonesia Nomor 16 Tahun 1997) Tentang Statistik. Jumlah petugas pencacah yang sangat terbatas di tiap Pangkalan Pendaratan Ikan (PPI), waktu untuk mengumpulkan dan kompilasi data jenis tangkapan dan berat sehingga terjadi keterlambatan hingga satu tahun untuk menyerahkan dan keakuratan data yang dikumpulkan.
- Penelitian-penelitian untuk mengetahui total penangkapan berat ikan masih dilakukan secara sampling dan ditimbang satu per satu secara manual (Triharyuni & Prisantoso, 2012).

Latar Belakang (lanjutan)

- Beberapa penelitian yang lain telah melakukan pengukuran ikan dengan menggunakan kamera digital (Shafry et al., 2012), (Anumudu & Mojekwu, 2015), (Jamaluddin et al., 2015), (Man et al., 2016), (Hao et al., 2017), (Sanchez-Torres et al., 2018), (Islamadina et al., 2018). Namun kelemahannya adalah menggunakan seting laboratorium dengan latar belakang yang sama dan jarak antara kamera dengan ikan yang tetap.
- Tantangan yang dihadapi adalah menggunakan kamera digital yang dipasang pada kapal nelayan dan *Convolutional Neural Network* (CNN) untuk dapat diujicobakan di lapangan yang sesungguhnya.

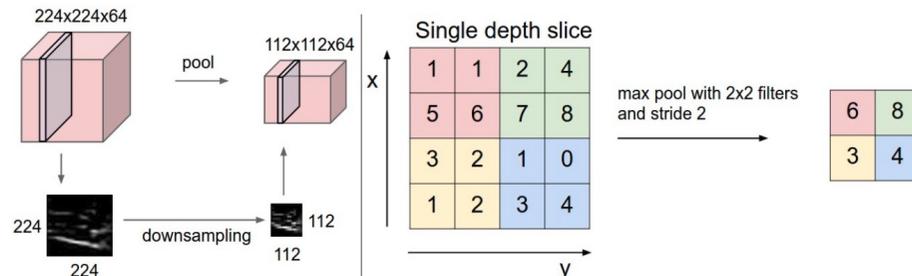
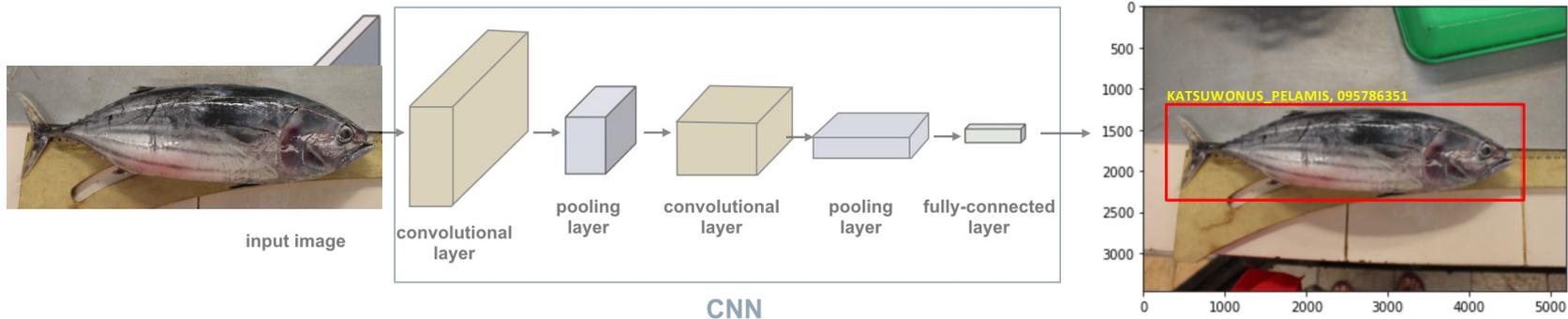
Latar Belakang (lanjutan)

- Di mana pergerakan ikan yang tertangkap sedang menggelepar di geladak dan sedang meluncur menuju ke lubang palka maka diperlukan sistem yang sudah dapat mengenali jenis ikan, panjang ikan dan beratnya.
- Seiring perkembangan CNN dengan jumlah lapisan (*layers*) yang semakin banyak maka proses pengenalan membutuhkan waktu proses yang semakin lama.

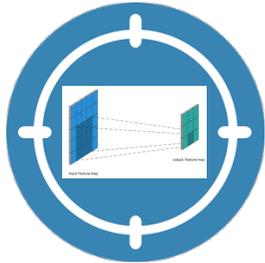


Latar Belakang (lanjutan)

- Dibutuhkan usaha untuk mempercepat proses training dan inferensi CNN dengan meningkatkan kinerja pada lapisan-lapisan tersebut. **MaxPool layer** adalah salah satu komponen utama pada arsitektur CNN yang menjadi fokus penelitian.



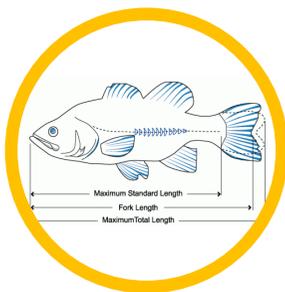
Perumusan Permasalahan



Bagaimana mengakselerasi algoritma *Maxpool2D*?



Bagaimana caranya mendeteksi obyek ikan dari sebuah gambar dengan latar belakang yang berbeda-beda?



Bagaimana mendapatkan panjang ikan yang sebenarnya dari hasil pendeteksian obyek?

Ruang Lingkup Penelitian

- ❑ Penelitian mengenai *Katsuwonus Pelamis* (Ikan Cakalang), *Euthynnus Affinis* (Ikan Tongkol), *Coryphaena Hippurus* (Ikan Lemadang) dan *Loligo Chinensis* (Cumi-cumi) menggunakan *Convolutional Neural Network (CNN)*.
- ❑ Algoritma YOLOV3-RESNET18
- ❑ Pada penelitian disertasi ini diset orientasi badan ikan adalah horisontal.



Tujuan Penelitian

1. Menemukan cara untuk mempercepat algoritma *Maxpool* dalam menghasilkan output
2. Mempelajari pendeteksian obyek dengan menggunakan *YOLOv3-ResNet18* agar dapat mengenali *Katsuwonus Pelamis* (Ikan Cakalang), *Euthynnus Affinis* (Ikan Tongkol), *Coryphaena Hippurus* (Ikan Lemadang) dan *Loligo Chinensis* (Cumi-cumi) dan dengan latar belakang yang berbeda-beda dan jarak yang statis.
3. Membangun sistem yang mendapat mengenali jenis ikan dan panjang ikan yang sebenarnya dari hasil pendeteksian obyek.

Kontribusi Penelitian

- Aspek keilmuan: peningkatan kecepatan proses *Convolution Neural Network* (CNN) pada *maxpool layer*. Pembuatan program *Python* yang dapat mengenali *Katsuwonus Pelamis* (Ikan Cakalang), *Euthynnus Affinis* (Ikan Tongkol), *Coryphaena Hippurus* (Ikan Lemadang) dan *Loligo Chinensis* (Cumi-cumi), mengetahui panjang ikan dan bobotnya dengan menggunakan kamera, yang sebelumnya dilakukan pengukuran dan penimbangan ikan secara manual dengan latar belakang yang berbeda-beda dan jarak yang statis.
- Aspek kontribusi praktis: program *Python* untuk mendapatkan panjang ikan yang sebenarnya hasil deteksi obyek dari sebuah gambar.

Tabel 2.3. Penelitian Terkait

No	Pengarang	Problem Domain	Metode	Output
1	Shafry, Rehman, Kumoi, Abdullah & Saba (2012) Man, Abdullah, Rahim & Amin (2016)	Mengukur panjang ikan	Teori optik dan teknik pemrosesan gambar. Menggunakan kamera digital 8 Megapixel. Jarak tetap dan latar belakang sama.	Akurasi pengukuran panjang ikan 95%
2	Anumudu & Mojekwu (2015)	Mengenali jenis (<i>species</i>) ikan yang memiliki bentuk yang mirip	Analisis morfometrik seperti <i>Truss Network Measurement</i> , analisis gambar dengan univariat, bivariat dan multivariat dan <i>Principal Component Analysis (PCA)</i> . Jarak tetap dan latar belakang sama.	Memperkenalkan teknik baru yang dapat dipergunakan untuk mengenali jenis ikan yang mirip lebih efisien $I \approx$
3	Jamaluddin, Seng, Shukor, Ibrahim, Miskon, Aras, Ghazaly, Ranom (2015)	Mengukur panjang ikan	Pendekatan pengukuran non-kontak menggunakan USB Camera, sensor jarak ultrasonik dan mikrokontroler. Jarak berbeda dan latar belakang sama.	Error 13.3% yang diakibatkan gelombang permukaan air

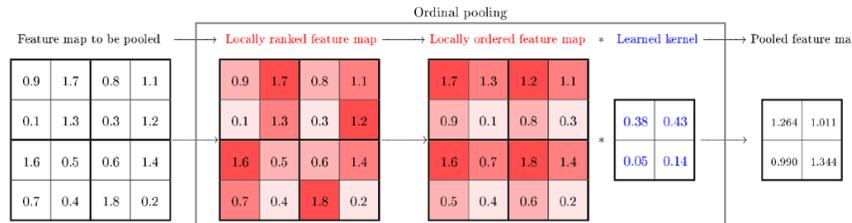
Penelitian Terkait (lanjutan)

4	Jin, Yan, Zhang, Fan (2015)	Hubungan berat dan Panjang ikan dan faktor kondisi <i>Fulton</i> ikan Cakalang	Hubungan berat dan Panjang ikan dan faktor kondisi Fulton ikan Cakalang	Hasil <i>fork length</i> yaitu 70% specimen ikan Cakalang di bawah 60 cm
5	Li, Shang, Qin, Chen (2015)	Mengenali ikan dengan menggunakan <i>Conventional Neural Network</i> (CNN)	Mengenali ikan dengan menggunakan CNN dan dataset ImageNet	Rata – rata presisi sebesar 81.4%
6	Hao, Yu, Li (2017)	Pengukuran panjang ikan	Visi Komputer dengan kamera tunggal dan kamera stereo. Pengukuran jarak dan perhitungan luas penampang ikan. Jarak dan latar belakang sama.	Error 5%

Penelitian Terkait (lanjutan)

7	Sanchez- Torres, Ceballos-Arroyo & Robles-Serrano (2018)	Pengukuran dan estimasi berat dan ukuran ikan	Pengolahan gambar penetasan ikan di bawah air. Regresi polinomial. Jarak berbeda. Latar belakang sama.	Error 11.88%
8	Islamadina, Pramita, Arnia, Munadi & Iqbal (2018)	Pengukuran badan ikan berupa estimasi panjang, lebar, dan tinggi	<i>Visual Capture</i> . Kamera tunggal. Jarak dan latar belakang sama.	Akurasi 80-95%
9	Cui, Zhou, Wang & Zhai (2020)	Pendeteksian ikan	Model CNN dengan segmentasi citra diperkenalkan untuk mendeteksi ikan dari dalam air laut yang buram.	Akurasi 83.2%
10	Ditria, Marcano, Sievers (2020)	Pendeteksian ikan	Menghasilkan 3 model dengan kerangka deteksi objek untuk mendeteksi target spesies, ikan yang penting secara ekologis dari video yang diolah per frame.	Akurasi 93.4%

Penelitian Terkait (lanjutan)

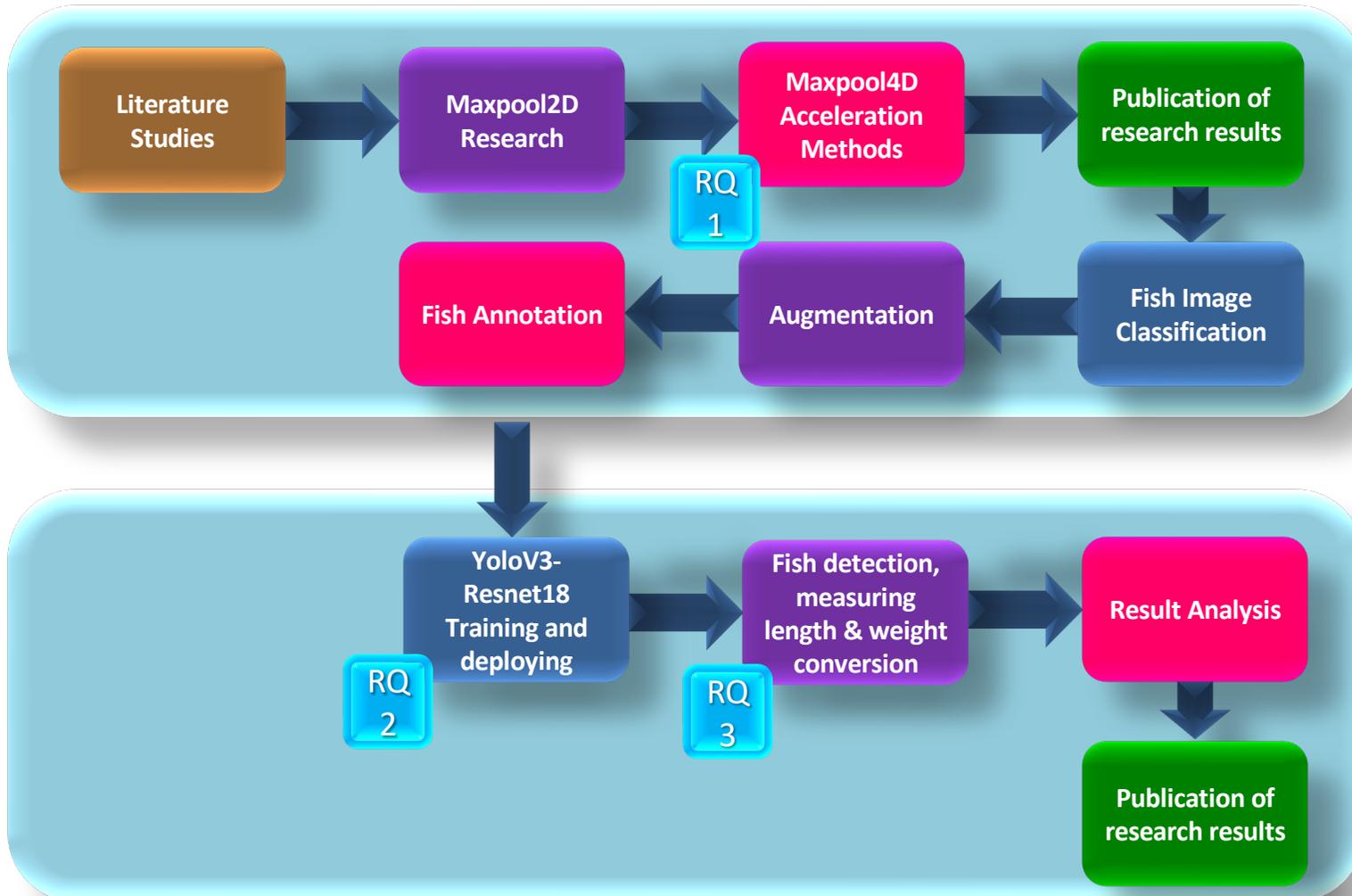
No	Metode Pooling	Keterangan
1	<i>Mixed Pooling</i>	Mixed pooling menggabungkan <i>Dropout</i> (Le et al., 2010) dan <i>Drop connect</i> (Zheng et al., 2014). Formulasi matematis dari <i>mixed pooling</i> diberikan sebagai berikut: Dimana, adalah bilangan acak yang mengacu pada <i>max pooling</i> atau <i>average pooling</i> . <i>Dropout</i> : membuat inisialisasi subset aktivasi yang dipilih secara acak menjadi nol. <i>Drop connect</i> : Alih-alih mengambil aktivasi yang dipilih secara acak, dibutuhkan bobot yang dipilih secara acak dan menginisialisasinya ke nol. Keduanya memiliki peran besar dalam regularisasi jaringan saraf. <i>Mixed pooling</i> memiliki kinerja yang lebih baik dibandingkan dengan rata-rata atau <i>max pooling</i> dan membantu dalam mengurangi masalah <i>overfitting</i> yang dihadapi oleh CNN.
2	<i>Stochastic Pooling</i>	Alih-alih mengambil pendekatan <i>pooling</i> deterministik (maksimum dan rata-rata), aktivasi diambil dari distribusi multinomial yang dibuat dengan aktivasi di wilayah <i>pooling</i> (Zeiler & Fergus, 2013). Menggunakan normalisasi aktivasi, <i>Stochastic Pooling</i> menghitung untuk setiap wilayah S_j , probabilitas P_i diberikan sebagai $P_i = \sum_{s \in S_i} (a_s)$. Pengambilan sampel dilakukan dari distribusi multinomial berdasarkan P_i untuk memilih lokasi k di dalam wilayah kecil, maka aktivasi kumpulan stokastik diberikan oleh $X_j = a_k$ dimana, $k \in P (p_1, p_2, \dots, p_j S_j)$
3	<i>Spectral Pooling</i>	<i>Fourier Spectral pooling</i> oleh Hubel pada tahun 1968 (Rippel et al., 2015) melakukan <i>down sampling</i> dari peta fitur input dengan menggunakan penyaringan lolos rendah dalam domain frekuensi. Metode ini memiliki kemampuan menyimpan informasi yang lebih baik dibandingkan dengan strategi pengumpulan spasial. Namun, itu tidak mengalami pengurangan tajam dalam dimensi keluaran.
4	<i>Ordinal Pooling</i>	Pengumpulan ordinal menggunakan konsep penyatuan ordinal untuk mengurangi efek buruk dari strategi penyatuan maks atau rata-rata (Deliège et al., 2020). Alih-alih mengambil nilai maksimum atau nilai rata-rata, ia mengatur ulang elemen dengan memberikan bobot yang berbeda ke elemen yang berbeda dan kemudian mengambil jumlah bobot elemen yang disusun ulang di wilayah penyatuan. Jenis penyatuan sangat penting dalam jaringan saraf dalam yang menggunakan operasi penyatuan berbeda dalam lapisan penyatuan yang sama. Ini juga meningkatkan akurasi di atas <i>maxpooling</i> atau rata-rata sambil mempercepat waktu pelatihan CNN.  <p>The diagram illustrates the Ordinal Pooling process. It starts with a 'Feature map to be pooled' (a 4x4 grid of values: 0.9, 1.7, 0.8, 1.1; 0.1, 1.3, 0.3, 1.2; 1.6, 0.5, 0.6, 1.4; 0.7, 0.4, 1.8, 0.2). This is transformed into a 'Locally ranked feature map' (a 4x4 grid where values are sorted within each row: 0.9, 1.7, 0.8, 1.1; 0.1, 1.3, 0.3, 1.2; 1.6, 0.5, 0.6, 1.4; 0.7, 0.4, 1.8, 0.2). This is then transformed into a 'Locally ordered feature map' (a 4x4 grid where values are sorted across the entire map: 1.7, 1.3, 1.2, 1.1; 0.9, 0.1, 0.8, 0.3; 1.6, 0.7, 1.8, 1.4; 0.5, 0.4, 0.6, 0.2). This is multiplied by a 'Learned kernel' (a 2x2 grid of values: 0.38, 0.43; 0.05, 0.14). The final result is a 'Pooled feature map' (a 2x2 grid of values: 1.264, 1.011; 0.990, 1.344).</p>

Habib, G., & Qureshi, S. (2020). Optimization and acceleration of convolutional neural networks: A survey. *Journal of King Saud University - Computer and Information Sciences*, in press.

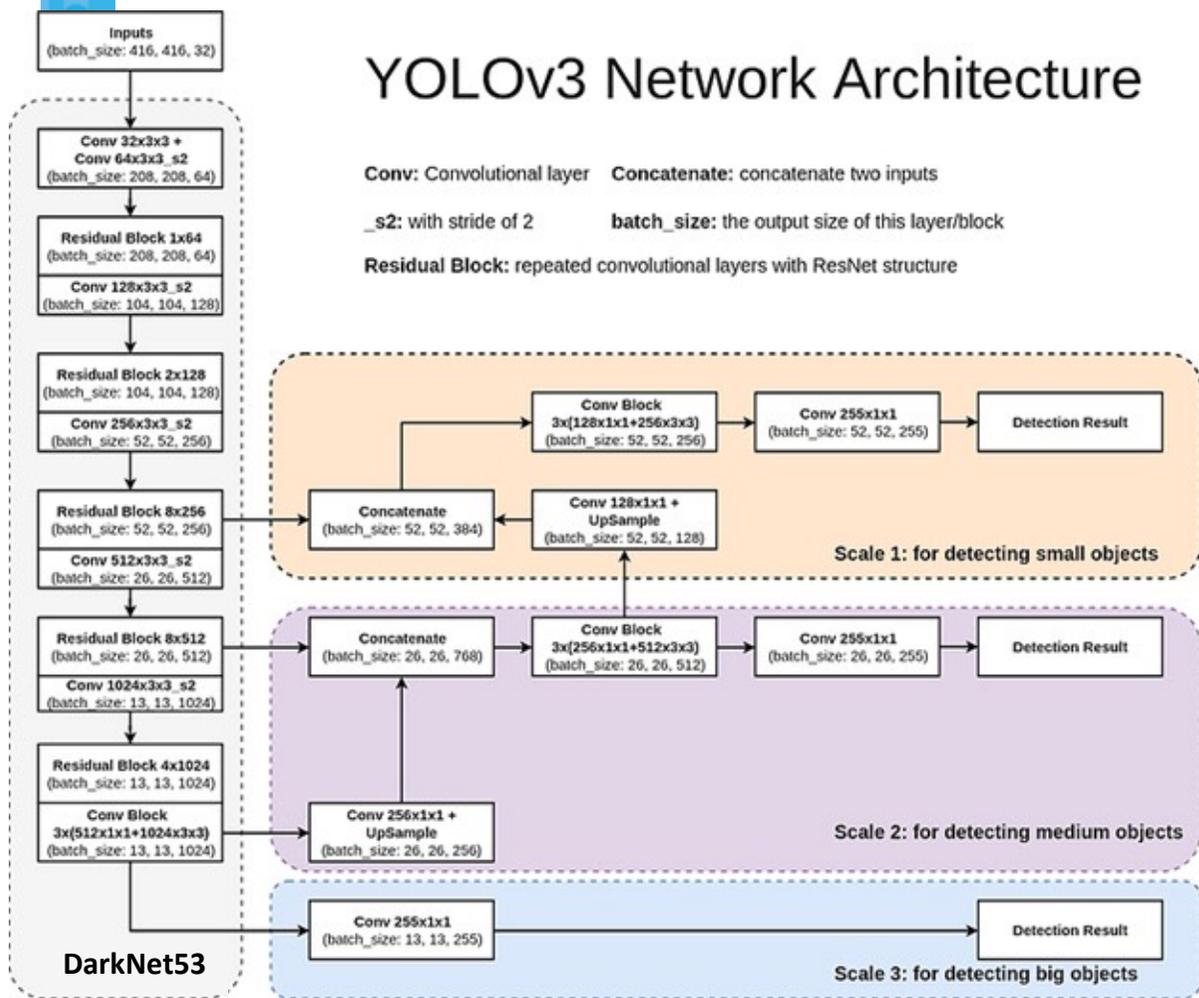
Penelitian Terkait (lanjutan)

- Dari penelitian-penelitian di atas maka dapat disimpulkan bahwa visi komputer dapat digunakan untuk mengenali jenis ikan dan mengukur panjang ikan menggunakan kamera baik tunggal atau stereo masih menggunakan latar belakang yang sama dan usaha untuk melakukan perubahan pada *convolution* dan ***maxpool layer*** masih memprosesnya secara berurutan satu persatu.
- Kontribusi penelitian ini pada latar belakang yang berbeda-beda, menggunakan dataset ikan sendiri (terdiri dari 4.000 foto), adanya estimasi bobot ikan berdasarkan panjang ikan yang terdeteksi berdasarkan ukuran dari *bounding box*nya, dan berupaya untuk melakukan peningkatan kecepatan *maxpool layer*.

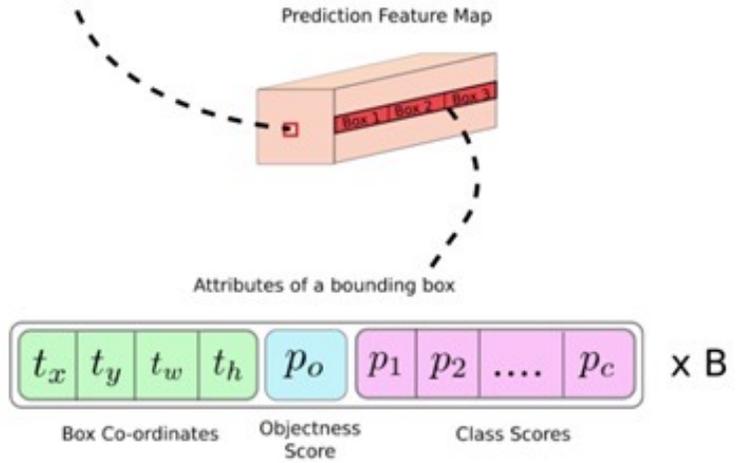
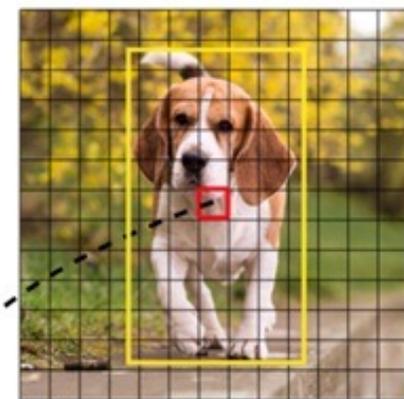
Kerangka Penelitian



YOLOv3 Network Architecture



Conv: Convolutional layer **Concatenate:** concatenate two inputs
 _s2: with stride of 2 **batch_size:** the output size of this layer/block
Residual Block: repeated convolutional layers with ResNet structure



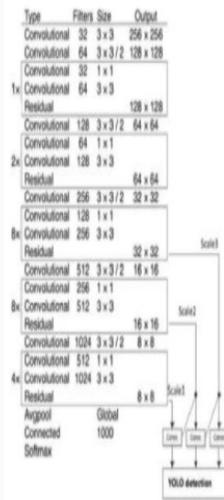
YOLOv3-ResNet18

Updated on August 31, 2020 09:45:34 Reading 1942 Reply 0 Show author Original poster

[Technical dry goods] Feature extraction network Resnet18 in the YOLOv3_Resnet18 algorithm

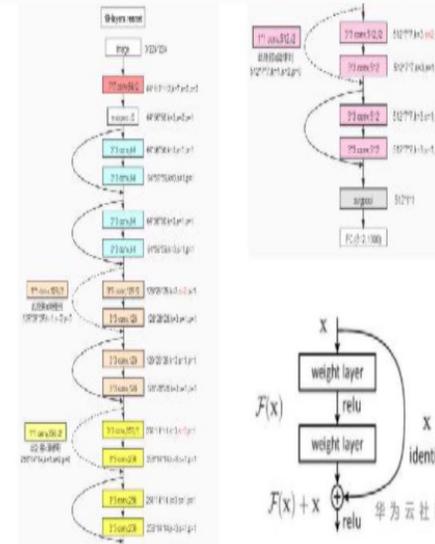
At present, the mainstream target detection algorithm YOLOv3 (of course, there are also YOLOv4), the feature extraction network has changed from Darknet53 to the classic Resnet18. The proposal of the Resnet network solves the problem that as the network deepens, the accuracy of the training set decreases, and the performance is improved.

Excellent, but compared to Darknet53, the feature extraction ability is a bit worse, but if you only do some simple detection, such as single-category detection with obvious characteristics, it is still possible.



使用了经典的Resnet网络, 选取18层深度, 既降低了计算量;

又由于昇腾 310处理器对Resnet网络做了专门优化, 可以提升运行速度。



From Darknet53 to Resnet18

Program Maxpool2D

```

42     # Iterate through image
43     counter = 0
44     for y in range(imagePadded.shape[0]): # y is row
45         # Exit Convolution
46         if y > imagePadded.shape[0] - yKernShape:
47             break
48         # Only Convolve if y has gone down by the specified Strides
49         if y % strides == 0:
50             for x in range(imagePadded.shape[1]): # x is column
51                 # Go to next row once kernel is out of bounds
52                 if x > imagePadded.shape[1] - xKernShape: # shape[1] adalah x atau column
53                     break
54                 try:
55                     # Only Convolve if x has moved by the specified Strides
56                     if x % strides == 0:
57                         yTemp = int(y/strides)
58                         xTemp = int(x/strides)
59                         hout = (kernel * imagePadded[y: y + yKernShape, x: x + xKernShape]).sum()
60                         if wantDebug and 0:
61                             print ("Position temp[{}, {}] = {}".format(yTemp, xTemp, hout))
62                         output[yTemp, xTemp] = (kernel * imagePadded[y: y + yKernShape, x: x + xKernShape]).sum()
63                         counter = counter + 1
64                 except:
65                     break
66     # print ('Total loop convolve2D = {}'.format(counter))
67     print(f"Total loop convolve2D = {counter:,}")
68     return output

```

Program Maxpool4D

```

for maxpoolRow in range(0, totalRow, tileMaxpool):
    for maxpoolCol in range(0, totalCol, tileMaxpool):
        # Get a tileBlock consists 16x16
        lastPart = image[maxpoolRow:maxpoolRow + tileMaxpool, maxpoolCol:maxpoolCol + tileMaxpool]
        blr, blc = np.shape(lastPart) # get the shape of the lastPart
        if (blr != tileMaxpool) or (blc != tileMaxpool): # Check if we are the last block
            aBlock = np.zeros((tileMaxpool, tileMaxpool))
            aBlock[0:blr, 0:blc] = lastPart # copy the lastPart into 16x16
        else:
            aBlock = lastPart
        # We process each cell. 16 values are calculated at a time
        for cellRow in range(cellMaxpool):
            for cellCol in range(cellMaxpool):
                for row in range(filterMaxpool):
                    for col in range(filterMaxpool):
                        # Take oneCell, find itsMax
                        sourceRow = (row * filterMaxpool) + (cellMaxpool * cellRow)
                        sourceRow2 = (row * filterMaxpool) + (cellMaxpool * (cellRow+1))
                        sourceCol = (col * filterMaxpool) + (cellMaxpool * cellCol) # 0, 4, 8, 12
                        sourceCol2 = (col * filterMaxpool) + (cellMaxpool * (cellCol+1)) # 0, 4, 8, 12
                        targetRow = maxpoolRow//2 + (row * cellMaxpool + cellRow)
                        targetCol = maxpoolCol//2 + (col * cellMaxpool + cellCol)
                        itsMax = np.max(aBlock[sourceRow:sourceRow2, sourceCol:sourceCol2])
                        temp[targetRow, targetCol] = itsMax
                    m = m + 1

out2 = temp[0:Nh, 0:Nw]
print("\ntotal maxPool2dt5d = ", m)
print(out2.shape)
return out2

```

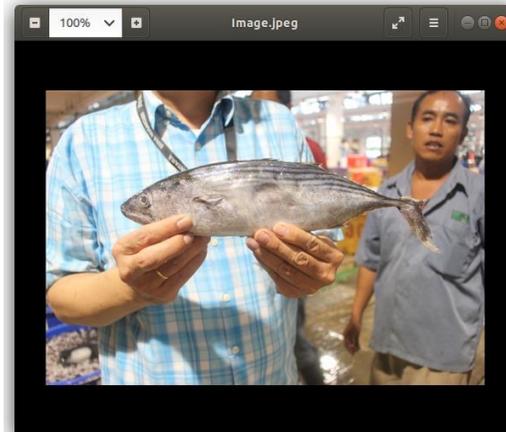


$O(n^2)$

maxpool2D							
Lebar x Tinggi	Resolusi (n)	y	x				Jumlah Loop maxpool2D
224x224	224	224	224				50,176
512x512	512	512	512				262,144
1024x1024	1,024	1,024	1,024				1,048,576
5120x5120	5,120	5,120	5,120				26,214,400
10240x10240	10,240	10,240	10,240				104,857,600

maxpool2DT4D								
Lebar x Tinggi	Resolusi (n)	maxpoolRow	maxpoolCol	cellRow	cellCol	row	col	Jumlah Loop maxpool2DTD
224x224	224	14	14	4	4	2	2	12,544
512x512	512	32	32	4	4	2	2	65,536
1024x1024	1,024	64	64	4	4	2	2	262,144
5120x5120	5,120	320	320	4	4	2	2	6,553,600
10240x10240	10,240	640	640	4	4	2	2	26,214,400

Hasil MaxPool



```
Run: main3 x
/usr/bin/python3.7 /home/suryadi/MaxPool/main3.py
Image size (WxH) = (416, 416)

total maxPool2d = 43264
(208, 208)
Process pool2d (filter4x4) = 267.03 miliseconds

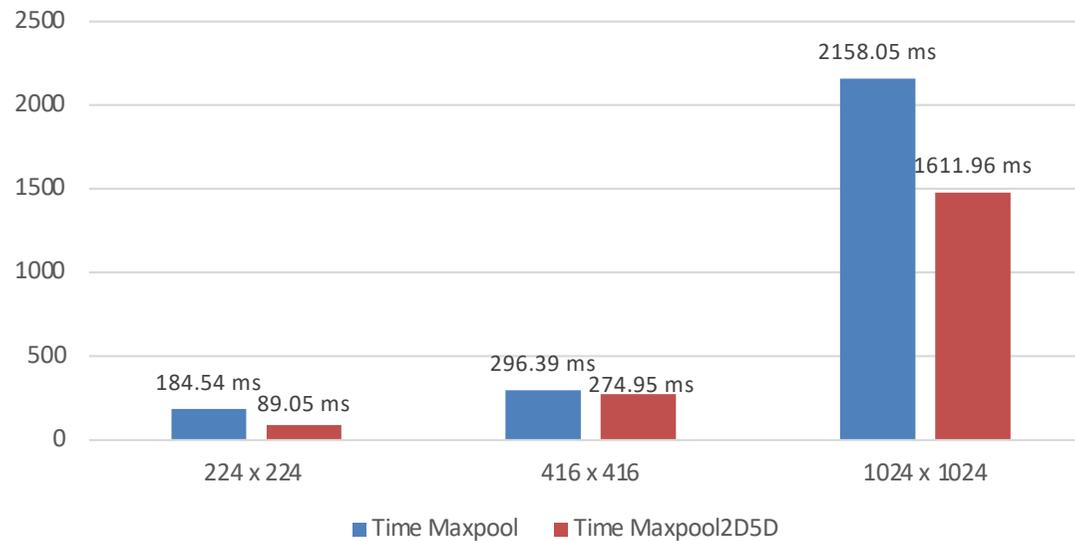
total maxPool2dt5d = 43264
(208, 208)
Process maxpool2dt5d (filter4x4) = 234.53 miliseconds
maxpool2dt5d faster = 32.50 miliseconds

Difference = [[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]

Process finished with exit code 0
```

Kinerja Maxpool 4D

Perbandingan *maxpool* dan *maxpool2DT4D*



Resolusi	<i>Maxpool</i> Tradisional	<i>Maxpool2DT4D</i>	Perbedaan	Persentase
224x224	184.54 ms	89.05 ms	95.49 ms	51.74%
416x416	296.39 ms	274.95 ms	21.44 ms	7.23%
1024x1024	2,158.05 ms	1,611.96 ms	546.09 ms	25.03%
		Rerata	221.01 ms	27.99%

Hasil Pengukuran

```
#show final result
plt.imshow(fin)
```

```
<matplotlib.image.AxesImage at 0x7f25e0111890>
```



Labeling Progress: 99%(4000/4002)



```
real_length = obj_length(focallength, imagewidth_pixels, sensorwidth_mm,
                        objectwidth_pixel,distance, score_flt, object_name)
formatted_length = [ '%.2f' % elem for elem in real_length]
real_weight = obj_weighth(real_length, ca, cb)
formatted_weighth = [ '%.2f' % elem for elem in real_weight]
```

Real KATSUWONUS_PELAMIS length (cm): 38.39

Real Weight (gr): 946.47

```
#save to local storage
cv.imwrite('output{}.jpg'.format(gambar), img_red)
```



Error estimasi panjang = 2,59%

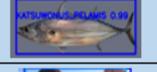
Error estimasi bobot = 15,67%

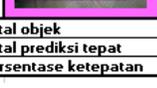
Hasil Pengukuran (lanjutan)

No	Output	Estimasi panjang (cm)	Selisih (cm)	Selisih %	Estimasi berat (gr)	Selisih (gr)	Selisih %
1		47.53	1.53	3.33	1667.63	257.63	18.27
2		46.83	0.83	1.80	1592.5	182.5	12.94
3		47.51	1.51	3.28	1665.28	255.28	18.10
4		46.77	0.77	1.67	1586.05	176.05	12.49
5		47.36	1.36	2.96	1648.84	238.84	16.94
6		47.52	1.52	3.30	1666.94	256.94	18.22
7		47.43	1.43	3.11	1656.34	246.34	17.47
8		46.98	0.98	2.13	1608.55	198.55	14.08
9		47.13	1.13	2.46	1623.92	213.92	15.17
10		46.84	0.84	1.83	1593.35	183.35	13.00
Rata-rata		47.19	1.19	2.59	1630.94	220.94	15.67

Hasil Pengukuran (lanjutan)

Pe
Im
Ex

No	Bounding Box Output	Hasil prediksi				
		Katsuwonus Pelamis	Euthynnus Affinis	Coryphaena Hippurus	Loligo Chinensis	Person
1		0.98	0	0	0	0
2		0.74	0	0	0	0.89
3		0.99, 0.98	0.32	0	0	0
4		0.89	0	0	0	0.94
5		0.99	0	0	0	0.98
6		0.71	0.31	0	0	0.99
7		0.42	0	0	0	0.95
8		0.98	0	0	0	0
9		0.59	0	0	0	0.58
10		0.95	0	0	0	0.95
Total objek		18				
Total prediksi tepat		17				
Persentase ketepatan		94.50%				

No	Bounding Box Output	Hasil prediksi				
		Katsuwonus Pelamis	Euthynnus Affinis	Coryphaena Hippurus	Loligo Chinensis	Person
1		0	0.97	0	0	0
2		0	0.91	0	0	0.99
3		0	0.96	0	0	0
4		0	0.91	0	0	0.92
5		0	0.99	0	0	0
6		0	0.96	0	0	0
7		0	0.88	0	0	0
8		0	0.97	0	0	0.86
9		0	0.65	0	0	0.52
10		0	0.94	0	0	0
Total objek		13				
Total prediksi tepat		13				
Persentase ketepatan		100.00%				

Hasil Pengukuran (lanjutan)

No	Bounding Box Output	Hasil prediksi				
		Katsuwonus Pelamis	Euthynnus Affinis	Coryphaena Hippurus	Loligo Chinensis	Person
1		0	0	0.95	0	0.98
2		0	0	0.9	0	0.97
3		0	0	0.83	0	0.94
4		0	0	0.93	0	0
5		0	0	0.98	0	0.94
6		0	0	0.98	0	0.99
7		0	0	0.97	0	0.89
8		0	0	0.53	0	0.89
9		0	0	0.98	0	0
10		0	0	0.89	0	0.9
Total objek		18				
Total prediksi tepat		18				
Persentase ketepatan		100.00%				

No	Bounding Box Output	Hasil prediksi				
		Katsuwonus Pelamis	Euthynnus Affinis	Coryphaena Hippurus	Loligo Chinensis	Person
1		0	0	0	0.98	0
2		0	0	0	0.97	0
3		0	0	0	0.88	0.92
4		0	0	0	0.94	0
5		0	0	0	0.9	0
6		0	0	0	0.99	0
7		0	0	0	0.95	0
8		0	0	0	0.98	0
9		0	0	0	0.95	0
10		0	0	0	0.97	0
Total objek		11				
Total prediksi tepat		11				
Persentase ketepatan		100.00%				

Akurasi Pengenalan

Tabel 4.8. Ketepatan Prediksi Tiap Kelas

No	Kelas	Akurasi (%)
1	Katsuwonus Pelamis	82.30
2	Euthynuss Affinis	91.40
3	Coryphaena Hippurus	89.40
4	Loligo Chinensis	95.10
Rerata		89.55

Hubungan Panjang dan Berat Ikan



Ikan Euthynnus Affinis (Pengolahan data sendiri)

Simpulan

- Untuk mempercepat proses pada maxpool layer adalah dengan mengurangi jumlah *loop* dan memperbanyak perhitungan hasil dalam setiap iterasinya. Dari hasil percobaan diperoleh percepatan *Maxpool2DT4D* rata-rata 221.01 ms (27.99%) dan kompleksitas maxpool dapat diturunkan menjadi 25% dari jumlah *loop* semula.

Simpulan

- Cara mendeteksi obyek ikan dari sebuah gambar dengan latar belakang yang berbeda-beda dan jarak yang statis adalah dengan melakukan training gambar *Katsuwonus pelamis* (ikan Cakalang), *Euthynnus affinis* (ikan Tongkol), *Coryphaena hippurus* (Lemadang) dan *Loligo Chinensis* (Cumi-Cumi) dengan *YOLOV3-Resnet18* dapat melakukan pendeteksian obyek dengan tingkat akurasi 89.55%.

Simpulan

- Perhitungan estimasi panjang objek yang ditangkap oleh kamera digital ditentukan oleh besarnya ukuran APS-C Sensor size dalam penelitian ini adalah 22.3 x 14.9 mm sehingga bila pada penerapan dengan menggunakan kamera lain dengan ukuran sensor yang berbeda dapat menyesuaikan dalam program perhitungannya. Penempatan kamera terhadap objek perlu secara perpendikular.

Simpulan

- Cara mengestimasi panjang obyek adalah dengan menemukan panjang obyek yang tertangkap pada sensor lensa kamera dalam pixel dengan mendeteksi *bounding box*-nya, jarak kamera dengan obyek yang diketahui, dan karakteristik lensa kamera yang digunakan maka panjang ikan yang sesungguhnya dapat dihitung. Dari hasil percobaan diperoleh error estimasi panjang sebesar 2.59% dan error estimasi berat sebesar 15.67%.

- Saran pengembangan yang dapat diberikan adalah melanjutkan penelitian dengan menerima input berupa video dan melakukan uji coba secara langsung dengan cara menempatkan kamera pada kapal nelayan, juga mengambil titik GPS kapal nelayan ketika melakukan penangkapan.
- Melakukan pengukuran panjang dan berat untuk mendapatkan garis regresinya
- Dengan berbasis cara kerja Maxpool2DT4D dapat diadaptasikan untuk melakukan modifikasi algoritma convolve2D.