

# KAEDAH AUGMENTASI IMEJ SINTETIK UNTUK PENGECAMAN BOT DAN KAPAL MENGGUNAKAN RANGKAIAN U-NET

Fakhrul Azran Bin Nawi dan Azizi Bin Abdullah

*Fakulti Teknologi & Sains Maklumat, Universiti Kebangsaan Malaysia*

## ABSTRAK

Pengesanan bot dan kapal dari kamera jarak jauh adalah salah satu tugas yang mencabar kerana saiz objek yang terlalu kecil jika diambil daripada pesawat. Sangat sukar untuk melihat dan mengenal pasti objek bot dan kapal kerana gangguan suasana persekitaran terutamanya daripada gelombang lautan dan pantulan cahaya matahari. Kepentingan untuk menentukan bot dan kapal adalah sangat diperlukan untuk tujuan keselamatan terutamanya jika kawasan adalah berdekatan dengan zon sempadan. Pada masa sekarang, keperluan seorang pakar yang berpengalaman adalah satu kemestian bagi menentukan sama ada objek itu adalah objek bot, kapal ataupun bukan. Cabaran utama dalam mengesan objek bot dan kapal adalah kerana saiz objek yang terlalu kecil jika menggunakan set data yang diambil dari pesawat seperti dron. Set data lain yang ada, yang berkenaan dengan bot dan kapal didapati tidak sesuai untuk kegunaan pemindahan pembelajaran kerana sudut objek yang tidak sama. Set data yang bersesuaian dengan kes penggunaan kajian ini sangat terhad dan telah merumitkan proses untuk mendapatkan model pembelajaran mendalam yang berketepatan tinggi. Teknik-teknik imej augmentasi yang sedia ada juga kurang efektif dengan kes penggunaan kajian ini. Kajian ini menggunakan set data Seagull kerana sudut pandangan dan jarak kamera yang hampir sama dengan kes penggunaan kajian ini. Walau bagaimanapun, set data ini sangat terhad dan ia mempengaruhi prestasi ketepatan model. Objektif kajian ini adalah untuk mengkaji penggunaan pembelajaran mendalam dengan menggunakan teknik imej segmentasi semantik iaitu U-Net bagi mengesan objek bot dan kapal. Selain daripada itu, kajian ini juga berjaya membangunkan dua kaedah imej augmentasi dengan menggunakan teknik imej sintetik yang dapat meningkatkan prestasi model pembelajaran mendalam. Teknik yang pertama memfokuskan kepada proses penempatan semula objek asal ke lokasi lain. Teknik yang kedua memfokuskan kepada proses penggandaan objek asal ke jumlah objek yang lebih banyak. Di akhir kajian ini, satu penilaian dan perbandingan telah dilaksanakan bagi mengukur tahap prestasi ketepatan model hasil daripada pembangunan kaedah imej augmentasi imej sintetik. Kajian ini memfokuskan penyelidikannya dengan menggunakan teknik segmentasi semantik sebagai seni bina utama bagi mengesan objek bot dan kapal. Kajian ini juga melakukan beberapa eksperimen berkenaan dengan penggunaan *hyper parameter* dan fungsi pengoptimum yang berbeza bagi mendapatkan prestasi model pembelajaran yang optimum. Hasil daripada kajian ini menunjukkan bahawa ketepatan *Dice Coefficient* telah meningkat daripada 70% kepada 77% dengan menggunakan imej sintetik penggandaan objek dan meningkat kepada 79% dengan menggunakan imej sintetik penempatan semula. Berdasarkan kepada hasil kajian, ia dapat disimpulkan bahawa terdapat perbezaan yang signifikan dalam menerapkan imej sintetik berbanding dengan hanya menggunakan set data yang asal.

## 1 PENGENALAN

Salah satu tanggungjawab agensi maritim Malaysia adalah untuk mengawal seluruh kawasan perairan Malaysia daripada pencerobohan luar. Selain daripada itu, agensi ini juga bertanggung jawab dalam menjalankan pengawasan udara dan lautan dan menjalankan tugas pencarian dan penyelamatan. Pada hari ini, mereka telah dilengkapi dengan sistem pemantauan pesawat. Namun begitu terdapat banyak lagi kerja-kerja operasi yang dilakukan secara manual yang boleh ditambah baik. Salah satu bidang yang boleh di tambah baik dalam operasi pemantauan adalah dengan mengadakan pengesanan awal informasi berkenaan dengan bot dan kapal. Imej

pada Rajah 1 menunjukkan bagaimana pemandangan laut diambil oleh agensi tersebut menggunakan pesawat drone.



Rajah 1 Pemandangan laut yang dirakam oleh agensi penguatkuasaan maritim

Dengan melihat menggunakan mata kasar, adalah sangat sukar bagi manusia untuk mengenali lokasi objek kapal yang berada di dalam imej. Cara yang paling mudah adalah dengan mengenal pasti ekor ombak yang dihasilkan daripada pergerakan kapal untuk menentukan sama ada objek itu adalah sebuah kapal atau bot. Walau bagaimanapun, terdapat banyak kapal di dalam imej yang tidak berada dalam kedudukan bergerak. Terdapat juga pelbagai jenis objek lain seperti perumahan dan pulau kecil yang boleh menyebabkan salah klasifikasi atau terlepas langsung daripada pengesanan. Cadangan dalam kajian ini adalah untuk memanfaatkan teknologi pembelajaran mendalam dalam pengesanan kapal dan bot. Terdapat banyak seni bina pembelajaran mendalam yang sedia ada dan kajian ini akan membandingkan beberapa teknik dalam menambak baik ketepatan seni bina itu.

Aplikasi pembelajaran mendalam dalam pengesanan objek telah popular sejak beberapa tahun kebelakangan ini bermula dengan terciptanya konvolusi rangkaian neural (CNN) oleh Lecun & Haffner (1999). Kemajuan pesat dalam bidang pengkomputeran membolehkan data diproses dengan lebih pantas. Unit pemprosesan grafik (GPU) bukan hanya memberikan enjin grafik yang kuat tetapi juga pemproses pengkomputeran yang berselari dan dapat mencapai kelajuan yang berterusan berbanding dengan CPU (Boyer & El Baz 2013).

Kerangka pembelajaran mesin seperti Tensorflow (Abadi et al. 2016), PyTorch (Ketkar 2017), Keras ("Keras API reference" t.th.), dan MXNet (Chen et al. 2015) membantu mempercepatkan perkembangan pembelajaran mesin dan juga dapat membantu mempercepatkan proses pembuatan pembelajaran mesin ke pengeluaran dengan lebih mudah.

Konsep baru di CNN seperti pemindahan pembelajaran (Hussain et al. 2019) dan model pra-latihan membantu mengurangkan jumlah data dan juga memendekkan masa yang diperlukan untuk latihan model CNN. Pemindahan pembelajaran adalah teknik peningkatan pembelajaran dalam tugas yang baru melalui proses pemindahan pengetahuan yang dihasilkan dari tugas yang telah dihasilkan dari pembelajaran mendalam yang lepas.

Pada hari ini, banyak aplikasi telah dibangunkan bagi membantu dalam menyelesaikan sesuatu masalah. Sebagai contoh dalam industri perubatan, kejuruteraan, dan pembangunan. Walaubagaimanapun, masih terdapat bidang industri yang masih menggunakan cara konvensional untuk mengesan objek sebagai contoh dalam industri maritim. Kajian penyelidikan ini memfokuskan kepada pembelajaran mendalam untuk pengesanan objek kapal dan bot.

Terdapat beberapa seni bina pembelajaran mendalam yang boleh digunakan dalam pengesanan objek kecil. Kajian daripada Fan et al. (2019) telah menilai perbezaan prestasi di antara beberapa seni bina pembelajaran mendalam iaitu CFAR, FRCNN, dan U-Net dan mendapati bahawa seni bina U-Net memperolehi prestasi yang tertinggi dengan ketepatan dan *recall* melebihi 90%.

## **2      PENYATAAN MASALAH**

Salah satu faktor penting bagi mencapai prestasi pembelajaran mendalam yang baik adalah dengan menyediakan set data yang cukup dan berkualiti tinggi. Namun begitu, terdapat banyak situasi yang mana data ini tidak dapat disediakan dengan lengkap. Kemungkinan besar adalah kerana data-data ini tidak di simpan dalam jangka masa yang lama kerana saiz data yang besar. Selain daripada itu, data-data yang rutin yang di rekod namun ia tidak memberi manfaat penting dalam bahagian operasi akan dibuang dan ini merupakan proses biasa dalam penyelenggaraan teknologi maklumat. Untuk tujuan kajian ini permasalahan utama yang ditekankan adalah berkisarkan

1.      Set data yang terhad dan tidak cukup untuk tujuan latihan pembelajaran mendalam.  
Set data yang sangat terhad akan mengurangkan ketepatan pembelajaran mendalam. Berdasarkan kajian yang dijalankan oleh Sun et al. (2017), mereka mendapati bahawa prestasi rangkaian meningkat secara linear dengan catatan jumlah data latihan. Dengan kata lain, semakin banyak data semakin baik prestasi rangkaian.
2.      Teknik imej augmentasi yang sedia ada kurang efektif pada kes penggunaan pengesanan bot.

Teknik biasa imej augmentasi yang sedia ada lebih sesuai digunakan untuk objek yang lebih besar manakala kes penggunaan untuk kajian ini mempunyai saiz objek sangat kecil.

3. Sukar mendapatkan kaedah yang sesuai dalam penjanaan imej sintetik.  
Proses pembikinan imej sintetik kebiasaannya dilakukan secara manual dengan menggunakan perisian seperti *Adobe Photoshop* atau pun perisian *Blender*. Proses ini merumitkan kerja-kerja penyediaan set data dan memakan masa.
4. Kesukaran untuk menentukan jenis seni-bina yang sesuai dalam pengesanan objek yang bersaiz kecil.  
Pengesanan objek yang bersaiz kecil di dalam imej mempunyai cabaran-cabaran yang berbeza berbanding objek-objek yang lebih besar. Kajian ini memfokuskan pada U-Net berdasarkan seni binanya yang mempunyai proses pensampelan ke bawah dan pensampelan ke atas dan diakhiri dengan penentuan klasifikasi objek pada peringkat piksel.

### **3 OBJEKTIF KAJIAN**

Berdasarkan pernyataan masalah, berikut adalah objektif utama dalam kajian ini

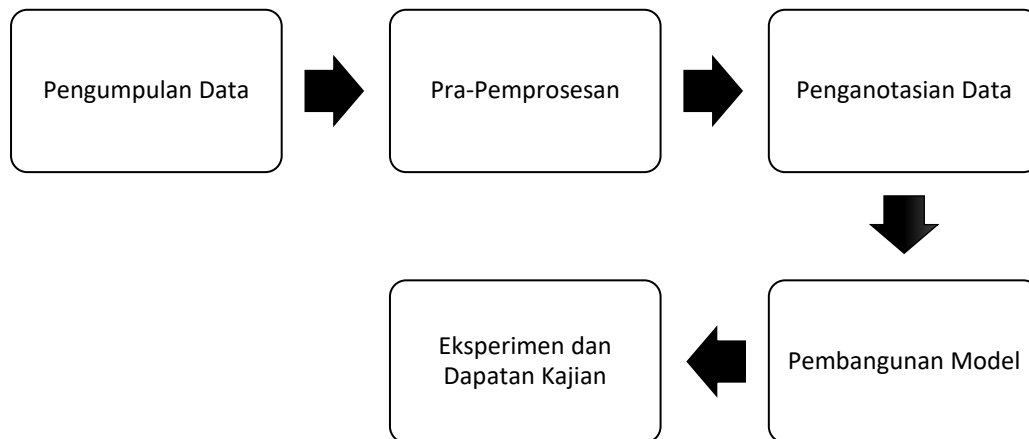
1. Mengkaji model pembelajaran mendalam untuk tujuan semantik segmentasi dengan menggunakan seni bina U-NET.
2. Membangunkan kaedah penjanaan imej sintetik.
3. Menilai prestasi keputusan dan dapatan kajian.

### **4 METODOLOGI**

Secara umumnya, kaedah-kaedah yang terlibat dipecahkan kepada beberapa proses penting. Proses-proses tersebut bermula dengan pengumpulan data, pra pemprosesan, pengelasan, latihan, dan analisis. Bagi memastikan kajian ini berjalan dengan lancar, penyediaan peralatan perkakasan dan perisian juga sangat penting. Kajian yang memfokuskan kepada pemprosesan imej memerlukan komputer berkuasa tinggi kerana setiap bit piksel adalah merupakan ciri-ciri yang akan diguna pakai dalam pembelajaran mendalam. Unit pemprosesan grafik (GPU) adalah merupakan satu keperluan dalam pemprosesan imej kerana keupayaannya untuk memproses setiap piksel secara serentak dalam satu masa yang sama. Ini membolehkan kajian dapat dilakukan dengan lebih pantas.

#### 4.1 Kerangka Aliran Proses

Kerangka aliran proses menerangkan secara terperinci langkah-langkah yang berlaku semasa pelaksanaan kajian ini. Langkah-langkah yang ditunjukkan bermula dengan pengumpulan data dan berakhir dengan proses kajian dan dapatan. Setiap aliran proses yang utama yang terlibat dalam kajian ini ditunjukkan pada Rajah 2.



Rajah 2 Kerangka aliran proses

Perjalanan kajian ini bermula dengan pengumpulan data. Set data yang digunakan di dalam kajian ini adalah dari sumber Seagull (“Seagull Dataset – VisLab – Computer and Robot Vision Laboratory” t.th.) dalam bentuk video. Langkah seterusnya ialah pra-pemprosesan yang mana set data dalam bentuk video diekstrak ke bentuk imej. Imej-imej yang diekstrak kemudian di semak dan imej yang tidak sesuai akan di asingkan. Kemudian, langkah penganotasi dijalankan untuk menganotasi objek bot dan kapal. Proses ini adalah proses pelabelan objek di dalam gambar untuk menentukan lokasi dan bentuk objek. Set data akhir hasil daripada proses anotasi kemudian digunakan pada langkah seterusnya iaitu langkah pembangunan model. Setelah selesai pembangunan model, model yang terhasil akan di nilai dan kajian dapatan di lakukan pada langkah yang terakhir.

#### 4.2 Pengumpulan Data

Pengumpulan data adalah proses mendapatkan set data untuk digunakan dalam pembangunan pembelajaran mendalam. Bagi kajian ini, set data yang digunakan diambil daripada sumber set data Seagull (“Seagull Dataset – VisLab – Computer and Robot Vision Laboratory” t.th.) iaitu

dalam format video. Sejumlah sepuluh video yang berdurasi di antara 10 saat sehingga 4 menit digunakan di dalam kajian ini. Set data ini merupakan kumpulan data multi spektrum dan multi-kamera yang disimpan bertujuan untuk penyelidikan pemantauan dan pengawasan laut. Set data ini dirakam dari sudut pandangan kenderaan udara tanpa pemandu (UAV) yang terbang di atas laut dan di atas kapal. Set data ini juga memberi beberapa jenis imej yang dirakam oleh kamera hyper spektrum namun untuk kajian ini hanya imej jenis RGB yang digunakan.

### **4.3 Pra-Pemprosesan**

Pra-pemprosesan merupakan proses awal yang penting dalam menjalankan kajian berkaitan pembelajaran mendalam. Proses ini adalah penting kerana pembelajaran mendalam akan menggunakan sepenuhnya setiap piksel di dalam gambar ini sebagai ciri-ciri kemasukan dalam pembelajaran. Set data yang kurang sempurna seperti tiada objek bot dan kapal, imej kabur, dan refleksi cahaya yang tinggi akan dikeluarkan daripada set data untuk latihan model.

### **4.4 Penganotasian Data**

Penganotasian data merupakan proses manual yang dilakukan oleh penyelidik untuk melabel atau menanda bentuk dan lokasi bot dan kapal yang berada di dalam imej. Untuk tujuan ini, kajian menggunakan perisian sumber terbuka iaitu *VGG Image Annotator* (VIA) (Dutta & Zisserman 2019) untuk menganotasikan imej.

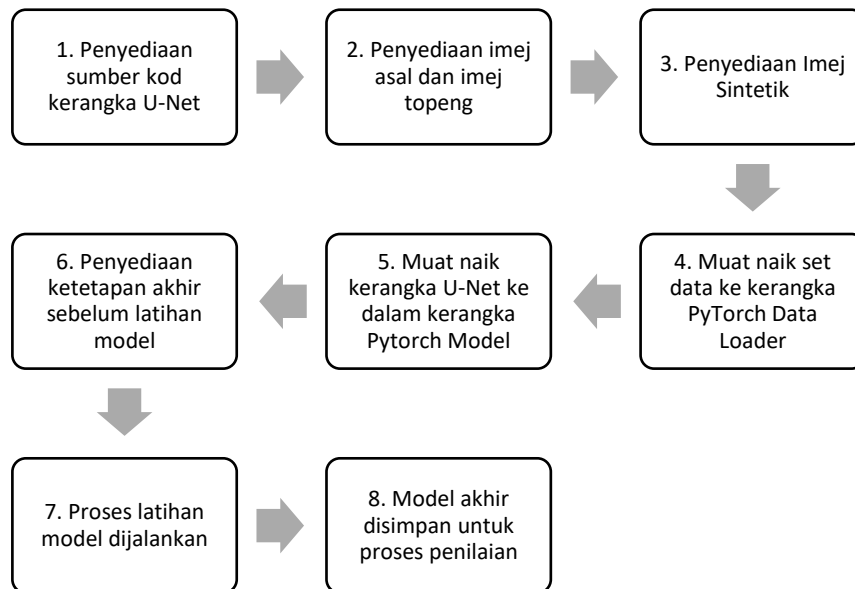
### **4.5 Pembangunan Model**

Pembangunan seni bina U-Net dalam kajian ini menggunakan kerangka pembangunan PyTorch. Sumber asal pengaturcaraan U-Net yang digunakan adalah berasal dari laman sesawang Smith, (“Creating and training a U-Net model with PyTorch for 2D & 3D semantic segmentation: Dataset building [1/4] | by Johannes Schmidt | Towards Data Science” t.th.). Penyelidik telah membangunkan seni bina U-Net ini untuk tujuan penyelidikannya dalam kajian electron tomograms iaitu dalam domain industri perubatan.

Untuk pembangunan pengaturcaraan seni bina U-Net yang dikongsi oleh Smith menggunakan set data dari Carvana (“Carvana Image Masking Challenge | Kaggle” t.th.) yang merupakan set data terkenal untuk imej segmentasi objek kereta.

Kerangka PyTorch telah menyediakan modul-modul yang memudahkan penyelidik dalam pembangunan pembelajaran mendalam. Modul penting yang mesti digunakan dalam penyediaan latihan model adalah modul Data Loader. Modul Data Loader membolehkan set data di muat naik dalam kerangka PyTorch sebelum proses latihan di mulakan.

Berikut adalah proses-proses penting yang digunakan dalam kajian ini bagi tujuan pembangunan model pembelajaran untuk pengesanan objek bot dan kapal. Rajah 3 menunjukkan carta aliran pembangunan model



Rajah 3 Carta aliran proses pembangunan model

#### 4.5.1 Penyediaan Sumber Kod Seni Bina UNET

Untuk kajian ini, sumber kod yang digunakan adalah berasal daripada kajian asal oleh Smith di laman sesawang (“johschmidt42/PyTorch-2D-3D-UNet-Tutorial” t.th.). Penyelidik telah menyediakan seni bina ini yang di kongsi di platform perkongsian GitHub. Namun begitu kod asal yang disediakan tidak mempunyai fungsi untuk menilai prestasi ketepatan model. Oleh yang demikian, fungsi penilaian ketepatan menggunakan pengiraan IoU dan Dice Coefficient ditambah ke dalam kod asal. Fungsi ini penting untuk membolehkan kajian ini menentukan aras prestasi sesuatu model.

Proses ini kemudiannya akan memuat turun kod akhir ke pelantar Kaggle daripada pelantar GitHub. Pelantar Kaggle merupakan pelantar yang akan digunakan untuk proses pembangunan model.

#### 4.5.2 Penyediaan Imej Asal dan Imej Topeng

Set data yang telah disiapkan pada proses penganotasian data digunakan sebagai sumber asas untuk pembangunan model pengesanan bot dan kapal. Sejumlah 300 keping gambar kesemuanya kemudian di muat naik ke pelantar Kaggle. Sejumlah 21 lagi set data untuk ujian

juga di muat naik ke pelantar yang sama untuk tujuan penilaian pada proses seterusnya. Setelah dimuat naik ke pelantar, set data ini perlu disediakan di dalam sesi pembelajaran model.

#### **4.5.3 Penyediaan Imej Sintetik**

Terdapat dua ujian yang memerlukan penyediaan imej sintetik dalam kajian ini. Secara amnya terdapat dua jenis penjanaan sintetik imej yang telah di hasilkan dalam kajian ini iaitu jenis penggandaan objek dan penempatan semula objek. Penerangan secara terperinci tentang penjanaan jenis sintetik imej ini akan diterangkan dalam Bab IV. Untuk ujian yang memerlukan imej sintetik, proses penyediaan imej sintetik ini merupakan satu keperluan. Proses ini dapat dijalankan dengan menggunakan skrip Python yang telah disediakan. Imej sintetik yang baru akan dijana, dihasilkan dan disimpan pada lokasi yang telah ditetapkan di dalam skrip Python itu. Imej yang dijana melalui proses akan digunakan pada proses yang seterusnya bergantung kepada jenis ujian kajian. Sebanyak 300 imej sintetik baru akan dijana untuk tujuan kajian ini.

#### **4.5.4 Muat Naik Set Data Ke Kerangka *Data Loader* Pytorch**

Set data yang telah tersedia akan di muat naik ke dalam kerangka Data Loader yang disediakan di dalam platform PyTorch. Set data asal sebanyak 300 keping imej akan di gunakan untuk tujuan latihan model. Untuk ujian yang menggunakan sintetik imej, penambahan 300 imej lagi akan digunakan dalam latihan model. Ini menjumlahkan keseluruhan set data berjumlah sebanyak 600 imej. Senyap 80 peratus daripada keseluruhan jumlah imej akan digunakan sebagai set data latihan dan selebihnya akan digunakan sebagai set data pengesahan.

#### **4.5.5 Muat Naik Seni Bina UNET Ke Pytorch**

Proses seterusnya adalah memuat naik seni bina U-Net ke dalam model PyTorch. Seni bina ini akan digunakan oleh PyTorch untuk proses latihan model. Parameter asas yang diperlukan untuk memuat naik seni bina U-Net adalah bilangan kemasukan saluran dan bilangan pengeluaran saluran. Untuk kajian ini, ia menggunakan kemasukan saluran sebanyak dua saluran yang mewakili jumlah saluran RGB imej dan pengeluaran sebanyak dua saluran yang mewakili objek bot dan latar belakang. Selain daripada itu, seni bina ini juga perlu dinyatakan jenis fungsi pengaktifan yang akan digunakan. Terdapat beberapa fungsi pengaktifan yang digunakan dalam kajian ini seperti contoh ReLU.



#### **4.5.6 Penyediaan Ketetapan Akhir Sebelum Latihan Model**

Proses ini adalah proses yang terakhir sebelum proses latihan model dijalankan. Proses ini adalah proses yang menentukan hyper parameter yang akan digunakan untuk ujian. Parameter yang digunakan adalah berbeza mengikut jenis ujian yang ingin dilakukan. Saiz yang digunakan di dalam latihan ini dikecilkan kepada dimensi 960 piksel x 536 piksel kerana keupayaan kuasa pemprosesan yang terhad. Jika saiz asal digunakan iaitu 1920 piksel x 1080 piksel akan menyebabkan unit pemprosesan kehilangan kesemua memori. Dalam proses ini juga akan ditentukan jumlah kadar pembelajaran yang akan digunakan sebagai contoh 0.001 ataupun 0.0001. Disini juga ditentukan jenis pengoptimum yang akan digunakan seperti Adam dan juga jenis pengiraan loss. Untuk kajian ini kesemua ujian menggunakan teknik pengiraan Cross Entropy Loss untuk pengiraan loss. Jumlah EPOCHS dalam setiap ujian juga ditetapkan pada kadar yang sama iaitu 50 EPOCHS untuk setiap ujian yang dijalankan.

#### **4.5.7 Proses Latihan Model**

Proses latihan model adalah proses utama yang mana pembelajaran model secara mendalam dilaksanakan. Proses ini adalah proses di mana rangkaian konvolusi neural akan menggunakan input yang dibekalkan, dan membandingkan dengan kelas yang sebenar, dan mengirakan pemberat dan nilai berat sebelah yang bersesuaian pada setiap nod. Proses ini akan berulang untuk setiap set data sebanyak 50 EPOCHS selesai. Setiap EPOCHS akan dikirakan nilai ketetapan model yang telah dibina.

#### **4.8 Eksperimen Dan Dapatan Kajian**

Hasil daripada ujian yang dijalankan kemudian di analisis pada proses ini. Terdapat tujuh siri ujian telah dilakukan kesemuanya yang mengambil kira penggunaan tetapan yang berbeza. Di antara tetapan yang di uji adalah penggunaan kadar pembelajaran (learning rate) yang berbeza dan penggunaan fungsi pengaktifan yang berbeza. Hasil tetapan yang optimum daripada ujian ini digunakan dalam ujian penggunaan imej sintetik.

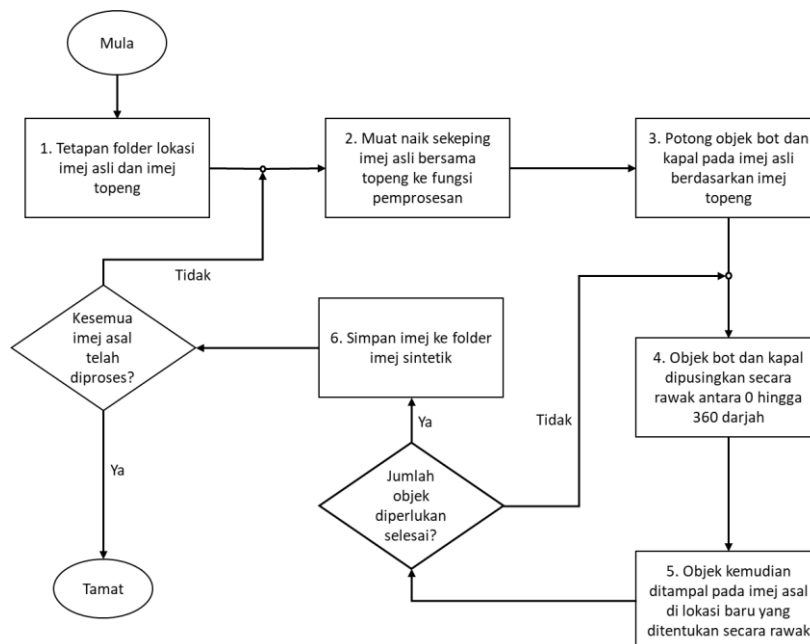
### **5 PEMBANGUNAN PENJANA IMEJ SINTETIK**

Untuk membolehkan kajian ini menggunakan imej sintetik secara teratur, penyelidik telah membina dua kaedah untuk menjana dua jenis sintetik imej berdasarkan imej asal. Dua jenis sintetik imej yang dijana adalah jenis penggandaan objek, dan penempatan semula objek. Kaedah ini dibangunkan dengan menggunakan pengaturcaraan Python dan kerangka OpenCV.

Imej augmentasi yang sedia ada secara amnya sesuai digunakan untuk kegunaan objek yang besar dan selalunya memenuhi hampir keseluruhan ruang gambar. Untuk kes penggunaan kajian ini, objek di dalam imej yang digunakan adalah sangat kecil yang merangkumi sebahagian peratusan sahaja daripada keseluruhan gambar.

## 5.1 Penggandaan Objek

Kaedah penggandaan objek merupakan satu fungsi pengaturcaraan yang bertujuan menggandakan jumlah objek di dalam imej. Jumlah objek asal pada imej asli hanya mempunyai satu objek sahaja. Teknik ini membolehkan objek asli itu di ulang mengikut jumlah tetapan yang diberikan. Dengan secara rawak fungsi ini meniru dan menampal imej ke lokasi baru manakala sudut orientasinya juga di pilih secara rawak. Proses-proses yang terdapat di dalam kaedah ini digambarkan pada Rajah 4.



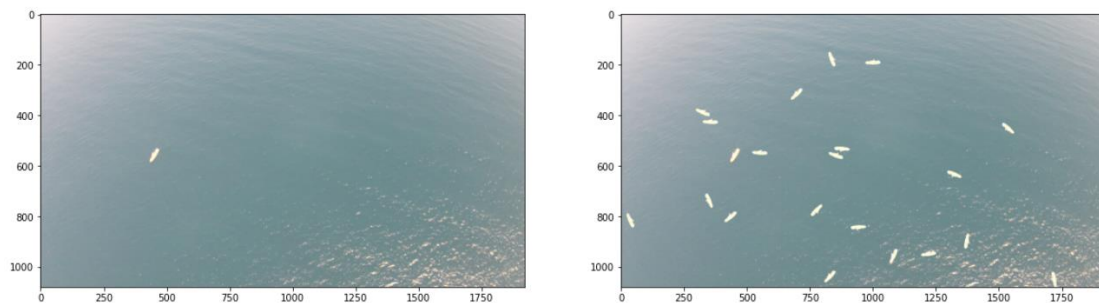
Rajah 4 Carta alir penjanaan imej sintetik bagi penggandaan objek

Berdasarkan carta aliran pada Rajah 4, kaedah penggandaan objek dijalankan mengikut langkah-langkah seperti berikut

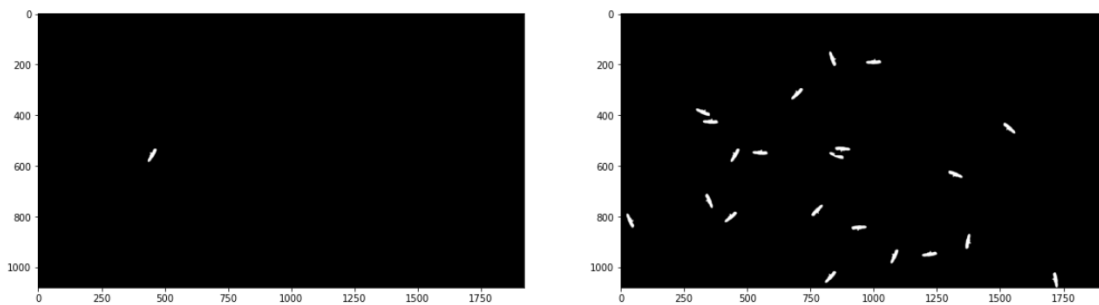
1. Langkah pertama dimulai dengan ketetapan lokasi folder di mana imej asli dan imej topeng disimpan.
2. Kemudian setiap pasang imej asli dan imej topeng akan di muat naik ke dalam fungsi pemrosesan. Untuk kajian ini, terdapat sebanyak 300 keping imej asli dan 300 keping imej topeng yang digunakan.

3. Objek pada imej asli itu kemudiannya di potong berdasarkan nilai yang terdapat pada imej topeng.
4. Objek yang telah di potong itu di pusingkan secara rawak diantara 0 hingga 360 darjah. Pusingan yang sama di lakukan pada objek di imej topeng.
5. Objek itu kemudiannya ditampal pada imej asal di lokasi yang baru yang juga di tentukan secara rawak.
6. Imej baru itu kemudian disimpan di lokasi folder imej sintetik.
7. Proses potong, pusing, dan tampal ini di ulang mengikut keperluan yang ditetapkan.

Untuk kajian ini, penyelidik menetapkan jumlah objek yang di tambah adalah sebanyak 20 mengikut kesesuaian saiz objek dan ruang keseluruhan imej. Rajah 5 dan Rajah 6 menunjukkan gambar imej asal berserta topeng dan imej sintetik berserta topeng yang telah dijana menggunakan teknik ini.



Rajah 5 Gambar menunjukkan imej asli di sebelah kiri dan imej sintetik di sebelah kanan

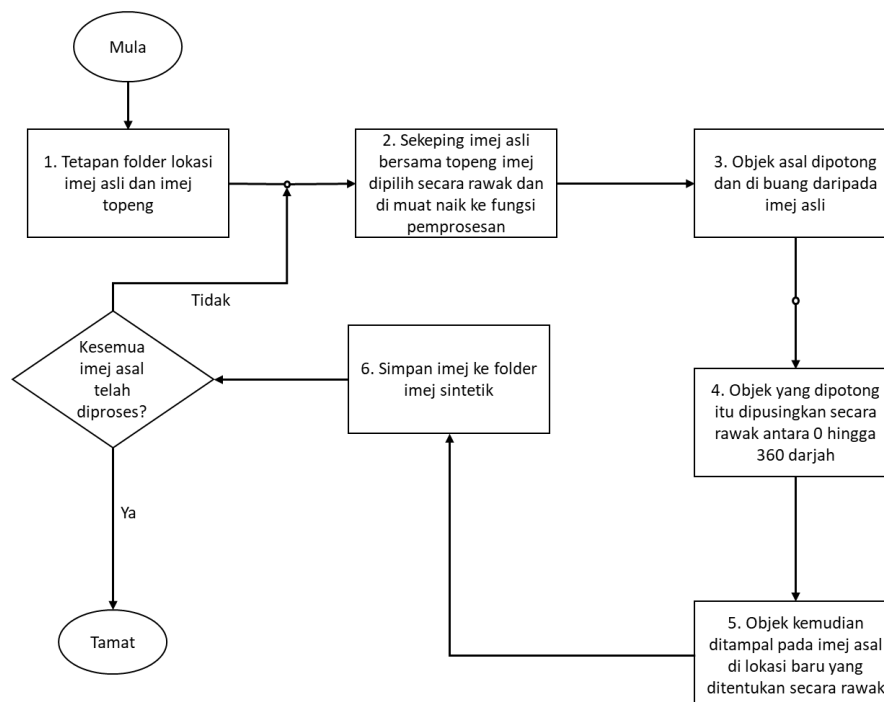


Rajah 6 Gambar menunjukkan imej topeng asli di sebelah kiri dan imej topeng sintetik di sebelah kanan

Setiap imej sintetik yang lengkap dijana akan disimpan ke satu folder imej sintetik untuk kegunaan pada proses latihan model pembelajaran mendalam. Keseluruhan proses ini diulang sehingga kesemua set data asal telah dijanakan dengan imej sintetik yang baru.

## 5.1 Penempatan Semula Objek

Kaedah penempatan semula objek merupakan satu fungsi pengaturcaraan yang menukar lokasi objek asal pada posisi lain. Latar belakang yang sama digunakan dalam proses ini. Proses-proses yang terdapat di dalam kaedah ini digambarkan pada Rajah 7.



Rajah 7 Carta alir penjanaan imej sintetik bagi penempatan semula objek

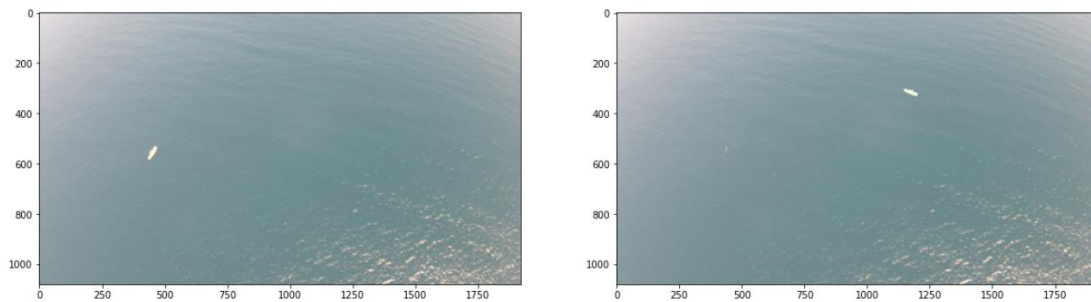
Berdasarkan carta aliran pada Rajah 7, kaedah penempatan semula objek dijalankan mengikut langkah-langkah seperti berikut

1. Kaedah ini bermula dengan ketetapan lokasi folder di mana imej asli dan imej topeng disimpan.
2. Kemudian setiap pasang imej asli dan imej topeng akan di muat naik ke dalam fungsi pemprosesan. Untuk kajian ini, terdapat sebanyak 300 keping imej asli dan 300 keping

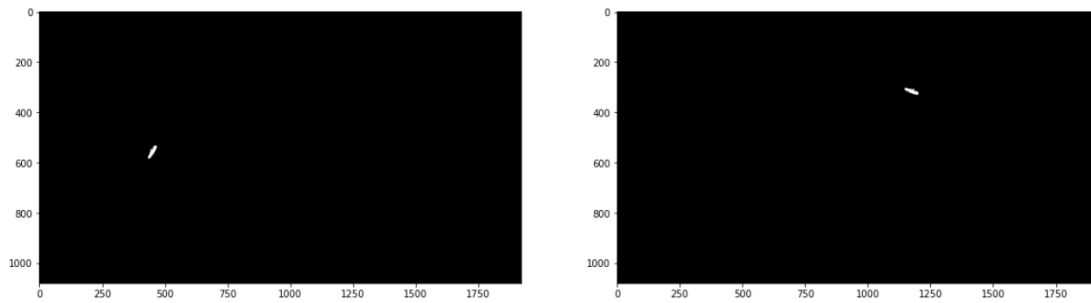
imej topeng yang digunakan. Pemilihan imej ke fungsi pemprosesan dijalankan secara rawak.

3. Objek pada setiap imej kemudiannya akan di potong dan di buang daripada imej asal menjadikan imej itu tidak mempunyai lagi objek bot dan kapal. Imej ini dikenali sebagai imej latar belakang.
4. Objek yang dipotong itu dipusingkan secara rawak diantara 0 hingga 360 darjah.
5. Objek yang telah dipusing kemudiannya di tampal pada lokasi baru pada imej latar belakang.
6. Imej baru itu kemudian disimpan di lokasi folder imej sintetik.
7. Keseluruhan proses ini diulang sehingga cukup 300 keping imej sintetik baru berjaya dijana..

Rajah 8 dan Rajah 9 menunjukkan gambar imej asal berserta topeng dan imej sintetik berserta topeng yang telah dijana menggunakan teknik ini.



Rajah 8 Gambar menunjukkan imej asli di sebelah kiri dan imej sintetik di sebelah kanan selepas proses penempatan semula



Rajah 9 Gambar menunjukkan imej topeng asli di sebelah kiri dan imej topeng sintetik di sebelah kanan selepas proses penempatan semula

## 6 HASIL KAJIAN

Terdapat beberapa siri ujian yang telah dilakukan dengan menggunakan beberapa kombinasi teknik dan parameter yang berlainan. Setiap siri ujian kemudiannya di nilai dengan menggunakan sukatan pengujian ketepatan dalam bentuk *Intersection of Union (IoU)* dan *Dice Coefficient*.

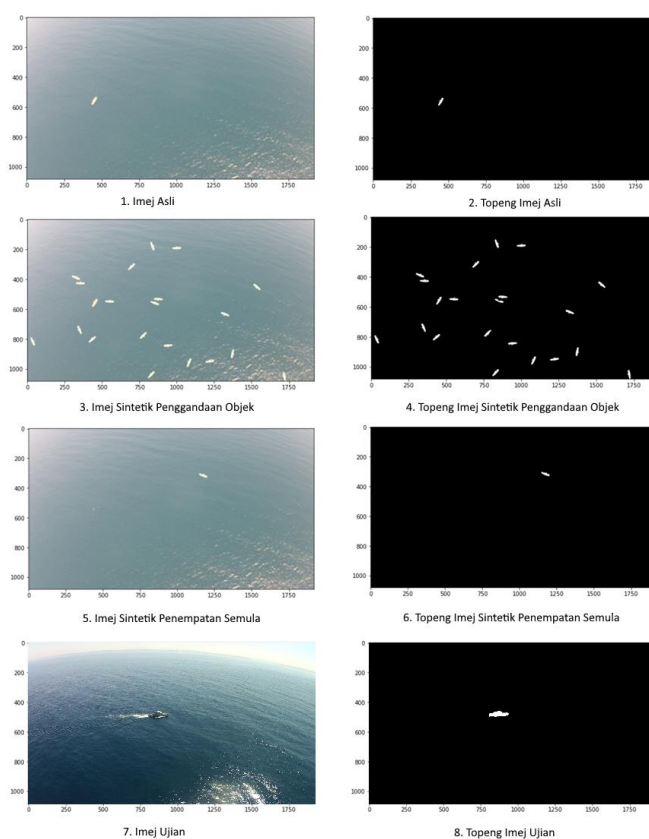
Untuk kajian ini, setiap saringan ujian yang dijalankan telah melalui proses pembelajaran sebanyak 50 EPOCHS. EPOCHS adalah bilangan saringan yang dilakukan oleh pembelajaran mendalam bagi menentukan pemberat dan yang sesuai. Untuk pengiraan kerugian (loss), kajian ini menyelaraskan kesemua ujian dengan menggunakan teknik Cross Entropy.

Sebanyak 300 imej asli bersama 300 imej topeng, 300 imej sintetik jenis penggandaan objek bersama 300 imej topeng, dan 300 imej sintetik jenis penempatan semula bersama 300 imej topeng, telah digunakan di dalam kajian ini. Setiap imej ini di bahagikan kepada set latihan dan set pengesahan dengan nisbah 80% : 20%. Nisbah yang digunakan ini merupakan nisbah yang sama yang telah diguna pakai oleh penyelidik asal. Ini menjumlahkan keseluruhan set data sebanyak 720 keping set imej untuk latihan dan 180 keping set imej untuk pengesahan. Pembahagian set latihan dan set pengesahan dipilih secara rawak. Kajian ini juga menggunakan 21 set imej asli dan 21 imej topeng yang digunakan untuk tujuan ujian yang mana ianya tidak melalui proses latihan pembelajaran mendalam. Jadual 1 menunjukkan senarai jumlah set data yang digunakan dalam kajian ini.

Jadual 1 Jumlah keseluruhan imej yang diguna dalam kajian ini

No	Jenis Imej	Jumlah Keseluruhan Imej	Jumlah Imej Untuk Latihan	Jumlah Imej Untuk Pengesanan
1	Imej Asli	300	240	60
2	Topeng Imej Asli	300	240	60
3	Imej Sintetik Penggandaan Objek	300	240	60
4	Topeng Imej Sintetik Penggandaan Objek	300	240	60
5	Imej Sintetik Penempatan Semula	300	240	60
6	Topeng Imej Sintetik Penempatan Semula	300	240	60
7	Imej Ujian	21	Tiada	Tiada
8	Topeng Imej Ujian	21	Tiada	Tiada

Rajah 10 menunjukkan contoh imej bagi setiap jenis di dalam Jadual 1.



Rajah 10 Gambar menunjukkan contoh imej bagi setiap jenis set data yang digunakan dalam kajian

## 6.1 Penentuan Kadar Pembelajaran Dalam Fasa Latihan

Ujian pertama adalah pengujian penggunaan kadar pembelajaran yang berbeza. Sebanyak dua saringan ujian telah dijalankan yang mewakili dua jenis penggunaan kadar pembelajaran yang berbeza iaitu dengan kadar 0.001 dan 0.0001. Ujian ini menggunakan set data imej asli sebanyak 300 keping imej berserta dengan 300 lagi imej topeng. Jadual 2 menunjukkan keputusan prestasi ketepatan bagi ujian kadar pembelajaran.

Jadual 2 Keputusan peratusan ketepatan bagi ujian kadar pembelajaran

Saringan	Jumlah Gambar Imej Asli	Jumlah Gambar Latihan	Jumlah Gambar Pengesahan	Fungsi Pengaktifan	Pengoptimum	Kadar Pembelajaran	Ketetapan IoU	Ketetapan Dice
1	300	240	60	SiLU	Adam	0.001	0.5708	0.7207
2	300	240	60	SiLU	Adam	0.0001	0.6757	0.8108

Berdasarkan hasil ujian, kadar pembelajaran pada saringan kedua iaitu dengan menggunakan kadar 0.0001 memberi hasil ketepatan yang lebih tinggi iaitu 68% bagi IoU dan 81% bagi *Dice Coefficient*. Sebaliknya ujian saringan pertama menunjukkan kadar ketepatan yang lebih rendah iaitu masing-masing dengan ketepatan sebanyak 60% dan 72% untuk IoU dan *Dice Coefficient*. Peningkatan ketepatan sebanyak 10% merupakan perbezaan yang amat ketara dalam pembelajaran mendalam.

## 6.2 Penentuan Kadar Penggunaan Fungsi Pengaktifan Dalam Fasa Latihan

Ujian kedua adalah pengujian penggunaan fungsi pengaktifan yang berbeza. Sebanyak tiga saringan telah dijalankan yang masing-masing menggunakan fungsi pengaktifan yang berbeza iaitu jenis Swish (SiLU), Leaky ReLU, dan ReLU digunakan untuk ujian ini. Jadual 3 menunjukkan keputusan prestasi ketepatan bagi ujian kadar penggunaan fungsi pengaktifan.

Jadual 3 Keputusan peratusan ketepatan bagi ujian fungsi pengaktifan

Saringan	Jumlah Gambar Imej Asli	Jumlah Gambar Latihan	Jumlah Gambar Pengesahan	Fungsi Pengaktifan	Pengoptimum	Kadar Pembelajaran	Ketetapan IoU	Ketetapan Dice
1	300	240	60	SiLU	Adam	0.0001	0.6757	0.8108
2	300	240	60	Leaky	Adam	0.0001	0.6801	0.8741
3	300	240	60	ReLU	Adam	0.0001	0.6506	0.8681



Berdasarkan hasil ujian, fungsi pengaktifan jenis Leaky ReLU memberi hasil ketepatan yang paling tinggi iaitu sebanyak 68% bagi IoU dan 87% bagi *Dice*. Ia disusuli dengan saringan tertinggi yang kedua iaitu jenis ReLU yang memberi ketepatan IoU sebanyak 65% dan *Dice* sebanyak 87%. Fungsi pengaktifan yang paling sedikit ketepatan ialah jenis SiLU iaitu sebanyak 68% bagi IoU dan 81% bagi *Dice*. Namun begitu jurang perbezaan diantara Leaky ReLU dan ReLU tidak jauh.

### 6.3 Ujian Penggunaan Imej Sintetik

Ujian ketiga iaitu ujian yang terakhir ini memberi fokus kepada penggunaan imej sintetik sebagai set data tambahan dalam pembelajaran mendalam. Terdapat dua jenis imej sintetik yang dijana iaitu jenis penggandaan objek dan jenis penempatan semula objek. Jenis fungsi pengaktifan dan pengoptimum diselaraskan menggunakan jenis ReLU dan Adam. Saringan pertama menggunakan sejumlah 300 imej asli yang digunakan untuk latihan dan pengesahan dalam latihan model. Sebanyak 21 imej lagi digunakan untuk ujian dan penentuan tahap prestasi ketepatan model. Saringan kedua pula menggunakan 300 imej asli dan 300 imej sintetik jenis penempatan semula berserta 21 imej lagi untuk kegunaan ujian. Saringan ketiga pula menggunakan 300 imej asli dan 300 imej sintetik jenis penggandaan objek berserta 21 imej lagi untuk kegunaan ujian. Jadual 4 menunjukkan keputusan prestasi ketepatan model latihan mengikut saringan yang telah dijalankan.

Jadual 4 Keputusan peratusan ketepatan bagi ujian imej sintetik

Sarin gan	Jenis Ujian	Jumlah Gambar	Jumlah Gambar Latihan	Jumlah Gambar Pengesahan	Jumlah Gambar Ujian	Kadar Pembela jaran	Ketetap an IoU	Ketetapan Dice
1	Penggunaan set data asli	300	240	60	21	0.0001	50%	70%
2	Penggunaan set data asli bersama imej sintetik penempatan	600	480	120	21	0.0001	61%	79%
3	Penggunaan set data asli bersama imej sintetik penggandaan	600	480	120	21	0.0001	61%	77%

Berdasarkan hasil ujian, penggunaan imej sintetik jenis penggandaan objek meningkatkan prestasi ketepatan sebanyak 11% bagi IoU dan sebanyak 7% bagi *Dice*

*Coefficient* yang membawa jumlah akhir ketepatannya iaitu sebanyak 61% dan 77% masing-masing untuk IoU dan *Dice Coefficient*. Manakala untuk penggunaan imej sintetik jenis penempatan semula objek, ia memberi keputusan peningkatan prestasi ketepatan sebanyak 11% bagi IoU dan 9% bagi *Dice Coefficient*.

## 7 KESIMPULAN

Secara kesimpulannya kajian ini telah mencapai tiga objektif utama kajian iaitu membangunkan kaedah untuk menjana imej sintetik, mengkaji teknik-teknik untuk meningkatkan prestasi ketepatan, dan membandingkan dan menganalisis hasil ujian.

Kajian ini telah menghasilkan dua kaedah penjanaan imej sintetik iaitu penjanaan penggandaan objek dan penempatan semula objek. Penjanaan imej sintetik jenis penggandaan objek adalah sangat disyorkan penggunaannya kerana ia membantu dalam meningkatkan prestasi ketepatan.

Penggunaan kadar pembelajaran pada kadar 0.0001 merupakan kadar yang terbaik untuk kes penggunaan pengesanan bot dan kapal. Jenis pengoptimum yang sesuai yang boleh digunakan adalah jenis ReLU dan Leaky ReLU.

## 8 RUJUKAN

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., et al. 2016. TensorFlow: A System for Large-Scale Machine Learning. *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, hlm. 265–283. Savannah, GA: {USENIX} Association. Retrieved from <https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi>
- Benchmark Suite – Cityscapes Dataset. (n.d.). <https://www.cityscapes-dataset.com/benchmarks/> [3 October 2021].
- Boyer, V. & El Baz, D. 2013. Recent advances on GPU computing in operations research. *Proceedings - IEEE 27th International Parallel and Distributed Processing Symposium Workshops and PhD Forum, IPDPSW 2013* 1778–1787. doi:10.1109/IPDPSW.2013.45
- Cane, T. & Ferryman, J. 2018. Evaluating deep semantic segmentation networks for object detection in maritime surveillance. *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, hlm. 1–6. doi:10.1109/AVSS.2018.8639077

- Carvana Image Masking Challenge | Kaggle. (n.d.). <https://www.kaggle.com/c/carvana-image-masking-challenge/data> [4 October 2021].
- Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., et al. 2015. MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems 1–6. Retrieved from <http://arxiv.org/abs/1512.01274>
- Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., et al. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Creating and training a U-Net model with PyTorch for 2D & 3D semantic segmentation: Dataset building [1/4] | by Johannes Schmidt | Towards Data Science. (n.d.). <https://towardsdatascience.com/creating-and-training-a-u-net-model-with-pytorch-for-2d-3d-semantic-segmentation-dataset-fb1f7f80fe55> [4 October 2021].
- Deng, W., Zheng, L., Ye, Q., Kang, G., Yang, Y. & Jiao, J. 2018. Image-Image Domain Adaptation With Preserved Self-Similarity and Domain-Dissimilarity for Person Re-Identification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- DeVries, T. & Taylor, G. W. 2017. Dataset Augmentation in Feature Space. *arXiv e-prints* arXiv:1702.05538.
- Dutta, A. & Zisserman, A. 2019. The {VIA} Annotation Software for Images, Audio and Video. *Proceedings of the 27th ACM International Conference on Multimedia, MM '19*. New York, NY, USA: ACM. doi:10.1145/3343031.3350535
- Emek, R. A. & Demir, N. 2020. Building detection from sar images using unet deep learning method. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives* 44(4/W3): 215–218. doi:10.5194/isprs-archives-XLIV-4-W3-2020-215-2020
- Everingham, M., Van~Gool, L., Williams, C. K. I., Winn, J. & Zisserman, A. (n.d.-a). The {PASCAL} {V}isual {O}bject {C}lasses {C}hallenge 2007 {(VOC2007)} {R}esults.
- Everingham, M., Van~Gool, L., Williams, C. K. I., Winn, J. & Zisserman, A. (n.d.-b). The {PASCAL} {V}isual {O}bject {C}lasses {C}hallenge 2012 {(VOC2012)} {R}esults.
- Fan, Q., Chen, F., Cheng, M., Lou, S., Xiao, R., Zhang, B., Wang, C., et al. 2019. Ship detection using a fully convolutional network with compact polarimetric SAR images. *Remote Sensing* 11(18). doi:10.3390/rs11182171
- Fritsch, J., Kühnl, T. & Geiger, A. 2013. A new performance measure and evaluation benchmark for road detection algorithms. *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, hlm. 1693–1700. doi:10.1109/ITSC.2013.6728473
- Gonzalez, T. F. 2007. Handbook of approximation algorithms and metaheuristics. *Handbook of Approximation Algorithms and Metaheuristics* 1–1432. doi:10.1201/9781420010749

- Goodfellow, I. J., Shlens, J. & Szegedy, C. 2014. Explaining and Harnessing Adversarial Examples. *arXiv e-prints* arXiv:1412.6572.
- He, K., Zhang, X., Ren, S. & Sun, J. 2015. Deep Residual Learning for Image Recognition. *CoRR* abs/1512.0. Retrieved from <http://arxiv.org/abs/1512.03385>
- Heaton, J. 2018. Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. *Genetic Programming and Evolvable Machines* 19(1): 305–307. doi:10.1007/s10710-017-9314-z
- Hussain, M., Bird, J. J. & Faria, D. R. 2019. A Study on CNN Transfer Learning for Image Classification. Dlm. Lotfi (pnyt.), Bouchachia (pnyt.), Gegov (pnyt.), Langensiepen (pnyt.), & McGinnity (pnyt.). *Advances in Computational Intelligence Systems*, hlm. 191–202. Cham: Springer International Publishing.
- johschmidt42/PyTorch-2D-3D-UNet-Tutorial. (n.d.).  
<https://github.com/johschmidt42/PyTorch-2D-3D-UNet-Tutorial> [4 October 2021].
- Kaggle: Your Home for Data Science. (n.d.). <https://www.kaggle.com/> [4 October 2021].
- Kampffmeyer, M., Salberg, A. B. & Jenssen, R. 2016. Semantic Segmentation of Small Objects and Modeling of Uncertainty in Urban Remote Sensing Images Using Deep Convolutional Neural Networks. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* 680–688. doi:10.1109/CVPRW.2016.90
- Keras API reference. (n.d.). <https://keras.io/api/> [3 October 2021].
- Ketkar, N. 2017. Introduction to PyTorch. *Deep Learning with Python: A Hands-on Introduction*, hlm. 195–208. Berkeley, CA: Apress. doi:10.1007/978-1-4842-2766-4\_12
- Krizhevsky, A., Sutskever, I. & Hinton, G. E. 2017. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* 60(6): 84–90. doi:10.1145/3065386
- Lecun, Y. & Haffner, P. 1999. Object Recognition with Gradient-Based Learning 319–345.
- Liu, Z., Gao, G., Sun, L. & Fang, Z. 2021. HRDNet: High-Resolution Detection Network for Small Objects 1–6. doi:10.1109/icme51207.2021.9428241
- Long, J., Shelhamer, E. & Darrell, T. 2014. Fully Convolutional Networks for Semantic Segmentation. *CoRR* abs/1411.4. Retrieved from <http://arxiv.org/abs/1411.4038>
- Maldonado-Bascon, S., Lafuente-Arroyo, S., Gil-Jimenez, P., Gomez-Moreno, H. & Lopez-Ferreras, F. 2007. Road-Sign Detection and Recognition Based on Support Vector Machines. *IEEE Transactions on Intelligent Transportation Systems* 8(2): 264–278. doi:10.1109/TITS.2007.895311
- Moon, N., Bullitt, E., van Leemput, K. & Gerig, G. 2002. Automatic Brain and Tumor Segmentation. Dlm. Dohi (pnyt.) & Kikinis (pnyt.). *Medical Image Computing and Computer-Assisted Intervention --- MICCAI 2002*, hlm. 372–379. Berlin, Heidelberg: Springer Berlin Heidelberg.

- Nalamati, M., Sharma, N., Saqib, M. & Blumenstein, M. 2020. Automated Monitoring in Maritime Video Surveillance System. *2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ)*, hlm. 1–6. doi:10.1109/IVCNZ51579.2020.9290533
- Ronneberger, O., Fischer, P. & Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *CoRR* abs/1505.0. Retrieved from <http://arxiv.org/abs/1505.04597>
- Seagull Dataset – VisLab – Computer and Robot Vision Laboratory. (n.d.). <https://vislab.isr.tecnico.ulisboa.pt/seagull-dataset/> [4 October 2021].
- Shorten, C. & Khoshgoftaar, T. M. 2019. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data* 6(1): 60. doi:10.1186/s40537-019-0197-0
- Simonyan, K. & Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* 1–14.
- Smith, B. 2018. U-Net Ship Detection in Satellite Optical Imagery. *U-Net Ship Detection in Satellite Optical Imagery*.
- Song, C., Xu, W., Wang, Z., Yu, S., Zeng, P. & Ju, Z. 2020. Analysis on the Impact of Data Augmentation on Target Recognition for UAV-Based Transmission Line Inspection. *Complexity* 2020. doi:10.1155/2020/3107450
- Štepec, D., Martinčič, T. & Skočaj, D. 2019. Automated System for Ship Detection from Medium Resolution Satellite Optical Imagery. *OCEANS 2019 MTS/IEEE SEATTLE*, hlm. 1–10. doi:10.23919/OCEANS40490.2019.8962707
- Sun, C., Shrivastava, A., Singh, S. & Gupta, A. 2017. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. *CoRR* abs/1707.0. Retrieved from <http://arxiv.org/abs/1707.02968>
- Szandala, T. 2020. Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks. *CoRR* abs/2010.0. Retrieved from <https://arxiv.org/abs/2010.09458>
- Tang, G., Liu, S., Fujino, I., Claramunt, C., Wang, Y., Tang, G., Liu, S., et al. 2021. H-YOLO : A Single-Shot Ship Detection Approach Based on Region of Interest Preselected Network To cite this version : HAL Id : hal-03096535 H-YOLO : A Single-Shot Ship Detection Approach Based on Region of Interest Preselected Network.
- Yang, X., Sun, H., Fu, K., Yang, J., Sun, X., Yan, M. & Guo, Z. 2018. Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation Dense Feature Pyramid Networks. *Remote Sensing* 10(1): 1–14. doi:10.3390/rs10010132
- Zhao, Z. Q., Zheng, P., Xu, S. T. & Wu, X. 2019. Object Detection with Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems* 30(11): 3212–3232. doi:10.1109/TNNLS.2018.2876865