



UNIVERSITAS INDONESIA

**Perbandingan Metode Dark Channel Prior dengan Metode Modified PDR-Net
untuk Single Image Dehazing**

Pengcitraan

Muhammad Hanif Fahreza	1906351026
Andre Septiano	1906398313
Muhammad Irfan Junaidi	1906293202
Muhammad Kenta Bisma Dewa Brata	1906350950

**FAKULTAS ILMU KOMPUTER
DEPOK**

Daftar Isi

Daftar Isi	2
Daftar Gambar	3
Daftar Tabel	4
1 Pendahuluan	5
1.1 Latar Belakang	5
1.2 Definisi Masalah	5
1.3 Tujuan Penelitian	6
2 Isi	7
2.1 Kajian Literatur	7
2.1.1 Metode Konvensional: <i>Dark Channel Prior</i>	7
2.1.2 Metode <i>Deep Learning: Modified</i> PDR-Net	15
2.2 Desain Eksperimen	21
2.3 Hasil Eksperimen	22
3 Penutup	25

3.1	Kesimpulan	25
3.2	Saran	25
3.3	Refleksi Kelompok	25

Daftar Gambar

2.1	Histogram intensitas <i>dark channel</i> piksel-piksel gambar <i>haze-free</i>	8
2.2	Histogram intensitas <i>dark channel</i> piksel-piksel gambar <i>hazy</i>	8
2.3	Ilustrasi persamaan (1) dan (2)	10
2.4	Contoh proses DCP	14
2.5	Arsitektur <i>Modified</i> PDR-Net	16
2.6	Arsitektur PDR-Net	16
2.7	Arsitektur FFA-Net	17
2.8	Arsitektur <i>Processing Module</i>	18
2.9	Arsitektur <i>Pre-Processing Module</i>	18
2.10	Arsitektur <i>Channel Attention Module</i>	19

Daftar Tabel

2.1	Perbandingan Restorasi Citra antara DCP dan MPDR-Net pada Dataset Test. . .	22
-----	---	----

2.2	Perbandingan Restorasi Citra antara DCP dan MPDR-Net pada Citra Kondisi Nyata.	23
2.3	Perbandingan SSIM dan <i>runtime</i> DCP dan MPDR-Net pada <i>testing set</i> MPDR-Net	23

Bab 1

Pendahuluan

1.1 Latar Belakang

Gambar-gambar yang diambil di luar ruangan biasanya akan mengalami penurunan kualitas yang diakibatkan oleh partikel-partikel yang bertebaran di atmosfer seperti kabut dan asap. Kabut dan asap atau biasa disebut dengan haze dapat mengubah warna dan kontras dari sebuah citra yang akan dapat mengurangi keterlihatan objek-objek yang ada pada suatu citra sehingga kualitas dari citra tersebut akan rusak.

Haze biasanya terjadi ketika ada penyerapan dan penyebaran partikel debu dan asap pada udara yang cenderung kering sehingga debu, asap, dan polutan lain menjadi berkumpul membentuk haze yang mengganggu pandangan kita pada citra. Maka dari itu, kita membutuhkan suatu cara untuk menghilangkan haze dari suatu citra.

Banyak metode yang sudah dibuat untuk menyelesaikan masalah ini. Mulai dari yang menggunakan teknik pengolahan citra yang relatif mudah dan simpel hingga menggunakan metode deep learning yang relatif kompleks. Pada laporan ini, kami memilih untuk melakukan percobaan dan analisis pada dua metode, yaitu: Dark Channel Prior for Single Image Haze Removal (DCP) yang diusulkan oleh He et al. (2010) dan Modified PDR-Net (MPDR-Net) yang diusulkan oleh Hartanto dan Rahadiani (2021).

1.2 Definisi Masalah

Berdasarkan latar belakang yang telah dipaparkan, rumusan permasalahan dalam penelitian ini adalah:

- Apakah metode DCP dapat melakukan *single image dehazing* pada citra yang *hazy*?
- Seberapa baik metode DCP dalam melakukan *single image dehazing* pada citra yang *hazy*?
- Apakah MPDR-Net dapat melakukan *single image dehazing* pada citra yang *hazy*?
- Seberapa baik MPDR-Net dalam melakukan *single image dehazing* pada citra yang *hazy*?

1.3 Tujuan Penelitian

Penelitian ini dilakukan dengan harapan dapat menjawab pertanyaan penelitian di atas. Hasil dari penelitian ini diharapkan dapat:

- Mengetahui seberapa baik metode DCP dalam melakukan *single image dehazing* pada citra yang *hazy*.
- Mengetahui seberapa baik MPDR-Net dalam melakukan *single image dehazing* pada citra yang *hazy*.

Bab 2

Isi

2.1 Kajian Literatur

2.1.1 Metode Konvensional: *Dark Channel Prior*

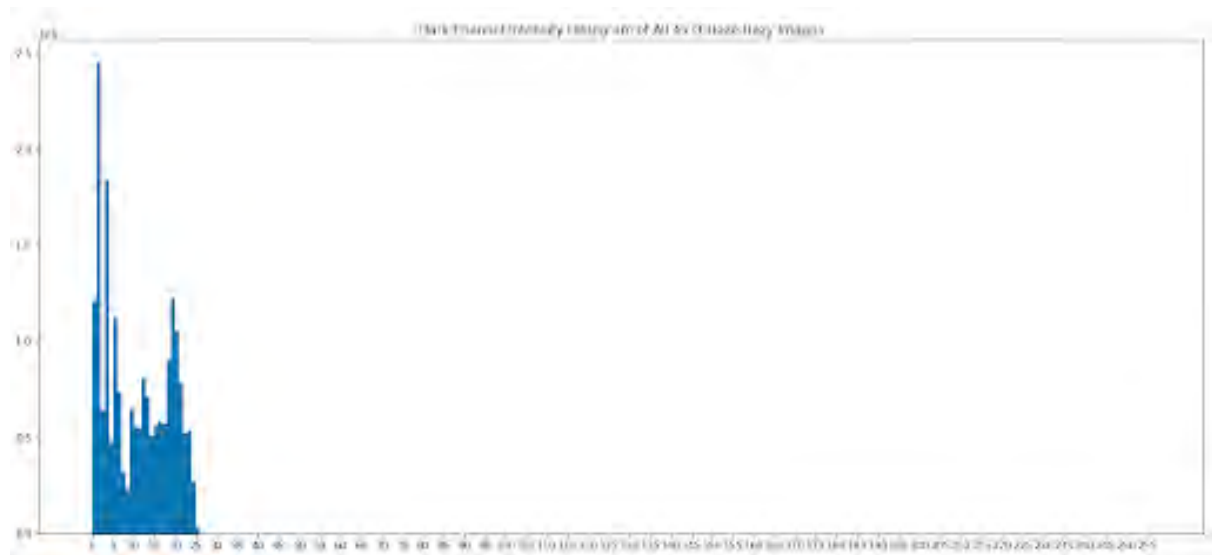
Metode *Dark Channel Prior* (akan kami sebut sebagai DCP pada laporan ini) adalah metode *haze removal* yang diusulkan oleh He et al. (2010). Sesuai namanya, metode ini mengandalkan sebuah asumsi atau i mengenai citra luar ruangan.

Berdasarkan observasi pada data statistik mengenai citra luar ruangan, kebanyakan bagian pada citra yang tidak menampilkan langit biasanya memiliki piksel-piksel dengan intensitas yang relatif sangat rendah pada setidaknya satu warna dari tiga warna pada kanal RGB. Piksel-piksel tersebut disebut sebagai “*dark pixels*” atau “piksel gelap”. Pada citra yang memiliki *haze*, biasanya piksel-piksel gelap ini mempunyai intensitas yang relatif lebih tinggi dikarenakan adanya *airlight* yang disebabkan oleh *haze*. Kita dapat menggunakan piksel gelap tersebut untuk mengestimasi transmisi atau besar cahaya yang masuk ke dalam medium pengambilan citra.

Kami melakukan sebuah eksperimen untuk melihat apakah asumsi ini memang benar. Pada eksperimen ini, kami membuat histogram intensitas *dark channel* piksel-piksel yang ada di 45 pasang gambar yang ada di dataset O-HAZE. Berikut adalah hasilnya:



Gambar 2.1: Histogram intensitas *dark channel* piksel-piksel gambar *haze-free*



Gambar 2.2: Histogram intensitas *dark channel* piksel-piksel gambar *hazy*

Dapat dilihat pada grafik pertama bahwa intensitas piksel gambar-gambar bebas *haze* dari dataset memang kebanyakan bernilai 0. Intensitasnya cenderung rendah dengan intensitas maksimal sekitar 16-17. Lalu dapat dilihat juga pada grafik kedua bahwa intensitas gambar-gambar *haze* juga cenderung rendah, tetapi meningkat dari sebelumnya dengan intensitas maksimal sekitar 25. Selain itu, intensitas kebanyakan piksel di gambar-gambar *haze* juga tidak lagi bernilai 0, melainkan sekitar 1-3. Hal ini menunjukkan bahwa asumsi DCP benar untuk dataset ini.

Pada bidang *computer vision* dan *computer graphics*, model yang sering dipakai untuk menggambarkan *haze* pada sebuah citra adalah:

$$\mathbf{I}(x) = \mathbf{J}(x)t(x) + \mathbf{A}(1 - t(x)) \quad (1)$$

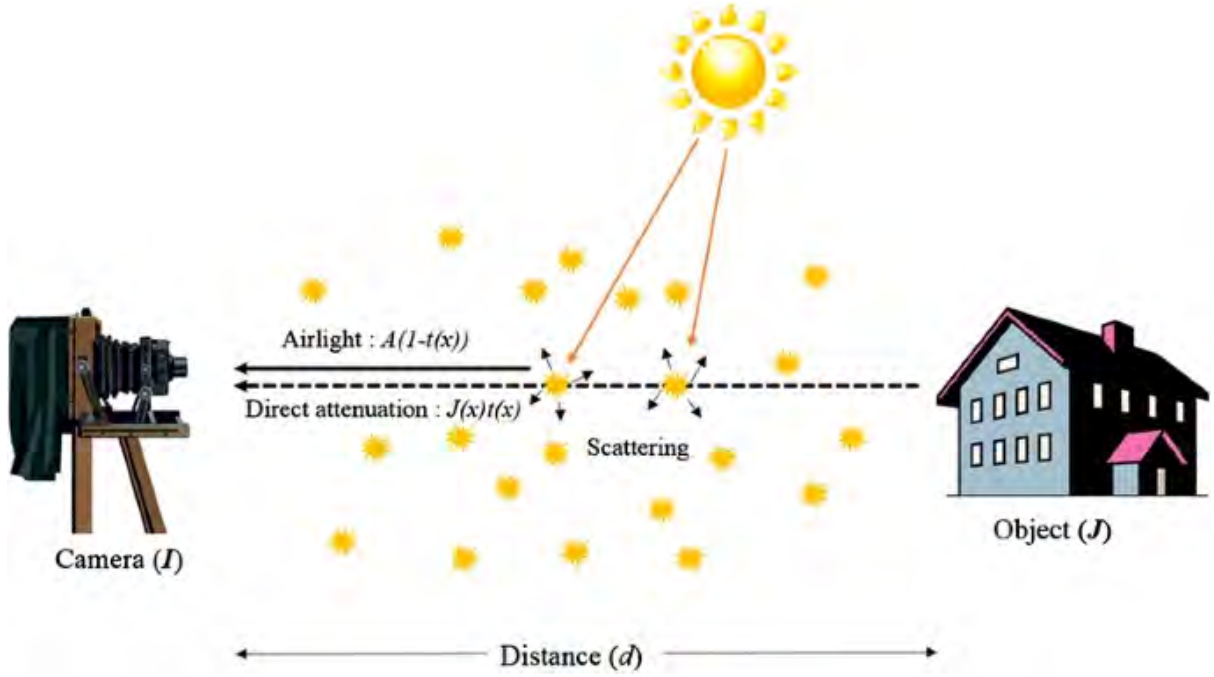
di mana $\mathbf{I}(x)$ adalah intensitas, \mathbf{J} adalah radiance citra bebas *haze*, \mathbf{A} adalah cahaya atmosfer global, dan t adalah transmisi medium, yaitu jumlah cahaya yang sampai ke medium pengambilan citra. Bagian $\mathbf{J}(x)t(x)$ dari persamaan disebut sebagai *direct attenuation* yang menyatakan seberapa banyak *radiance* dari citra yang berhasil mencapai medium pengambilan gambar. Bagian $\mathbf{A}(1 - t(x))$ adalah *airlight* yang disebabkan oleh keberadaan *haze* pada citra. Dengan menggunakan DCP, kita akan mengestimasi \mathbf{J} , \mathbf{A} , dan t yang sebelumnya tidak diketahui dari \mathbf{I} .

Fungsi transmisi $t(x)$ ketika atmosfer dianggap seragam adalah:

$$t(x) = e^{-\beta d(x)} \quad (2)$$

di mana β adalah koefisien hamburan udara pada atmosfer dan $d(x)$ adalah kedalaman suatu bagian dari citra. Dari persamaan tersebut, kita dapat melihat bahwa semakin dalam posisi objek yang ditampilkan oleh suatu citra, maka transmisi dari citra tersebut akan berkurang secara eksponensial.

Kedua persamaan diatas digambarkan dengan gambar di bawah.



Gambar 2.3: Ilustrasi persamaan (1) dan (2)

Dark channel dari suatu citra dapat dijabarkan secara formal dengan:

$$J^{dark}(x) = \min_{c \in \{r, g, b\}} \left(\min_{y \in \Omega(x)} J^c(y) \right) \quad (3)$$

dengan J^{dark} adalah dark channel, J^c adalah kanal warna dari J , dan $\Omega(x)$ adalah subcitra dengan titik tengah x . Berdasarkan asumsi DCP, intensitas dari J^{dark} pada bagian dari citra yang tidak menampilkan langit sangat rendah sampai-sampai cenderung bernilai 0. rendahnya intensitas *dark channel* ini dapat disebabkan oleh bayangan, benda-benda yang memang berwarna gelap, dan benda-benda yang mempunyai suatu warna yang dominan sehingga pasti ada salah satu kanal warna yang rendah pada citra benda tersebut. Dengan asumsi bahwa transmisi citra $\tilde{t}(x)$ pada suatu subcitra $\Omega(x)$ adalah konstan, kita dapat melakukan operasi $\min()$ pada persamaan 1 untuk subcitra $\Omega(x)$ menjadi:

$$\min_{y \in \Omega(x)} (I^c(y)) = \tilde{t}(x) \min_{y \in \Omega(x)} (J^c(y)) + (1 - \tilde{t}(x))A^c \quad (4)$$

Operasi \min tersebut kita lakukan untuk ketiga kanal RGB. Selanjutnya, kita lakukan operasi $\min()$ lagi untuk tiga hasil yang sudah kita dapatkan dari tiga operasi $\min()$ pada ketiga kanal RGB sebelumnya. Hasil persamaan yang didapat adalah:

$$\min_c \left(\min_{y \in \Omega(x)} \left(\frac{I^c(y)}{A^c} \right) \right) = \tilde{t}(x) \min_c \left(\min_{y \in \Omega(x)} \left(\frac{J^c(y)}{A^c} \right) \right) + (1 - \tilde{t}(x)) \quad (5)$$

Dengan menggunakan asumsi DCP, maka kita mengetahui bahwa *dark channel* atau J^{dark} dari *radiance* suatu citra bebas *haze* cenderung bernilai 0, sehingga kita dapat mengasumsikan:

$$J^{dark}(x) = \min_c \left(\min_{y \in \Omega(x)} (J^c(y)) \right) = 0 \quad (6)$$

Karena A^c selalu bernilai positif, maka kita dapat mengubah persamaan (6) menjadi:

$$\min_c \left(\min_{y \in \Omega(x)} \left(\frac{J^c(y)}{A^c} \right) \right) = 0 \quad (7)$$

Dengan menerapkan hasil yang kita dapatkan pada persamaan (7) pada persamaan (5), kita dapat melakukan estimasi nilai $\tilde{t}(x)$, yaitu:

$$\tilde{t}(x) = 1 - \min_c \left(\min_{y \in \Omega(x)} \left(\frac{I^c(y)}{A^c} \right) \right) \quad (8)$$

Kita dapat melihat bahwa bagian kanan persamaan tersebut adalah *dark channel* dari citra *haze* yang dinormalisasi oleh A^c .

Sebelumnya telah dinyatakan bahwa asumsi DCP hanya berlaku untuk bagian dari citra yang tidak menampilkan langit. Namun, karena warna dari langit biasanya sangat mirip dengan cahaya atmosfer A pada suatu citra *haze*, maka kita bisa menganggap bahwa:

$$\min_c \left(\min_{y \in \Omega(x)} \left(\frac{I^c(y)}{A^c} \right) \right) \rightarrow 1, \text{ and } \tilde{t}(x) \rightarrow 0 \quad (9)$$

Karena itu persamaan (8) juga berlaku untuk bagian langit dari citra sehingga kita tidak perlu melakukan pemisahan bagian langit dan bagian lain dari citra.

Walaupun *haze* dianggap merusak citra, kita tetap membutuhkan sedikit *haze* pada citra karena *haze* merupakan hal yang penting bagi manusia untuk dapat melihat kedalaman dari suatu pemandangan atau citra. Oleh karena itu konstan ω diperkenalkan pada persamaan (8) untuk menyimpan sedikit *haze* dari citra. Maka, persamaan tersebut menjadi:

$$\tilde{t}(x) = 1 - \omega \min_c \left(\min_{y \in \Omega(x)} \left(\frac{I^c(y)}{A^c} \right) \right). \quad (10)$$

He menetapkan $\omega = 0.95$.

Hasil estimasi $t(x)$ yang didapatkan dari persamaan (10) masih terlihat kasar sehingga perlu diperhalus. He menggunakan algoritma *soft matting* yang diusulkan oleh Levin et al. (2008) untuk memperhalus estimasi transmisi yang didapatkan. Namun, algoritma itu cukup sulit dan rumit untuk diimplementasikan. Karena itu He et al. (2013) mengusulkan suatu metode baru untuk menggantikan algoritma tersebut, yaitu metode *Guided Filter* (GF). Maka pada laporan ini kami akan menggunakan GF untuk memperhalus transmisi yang didapat.

GF mengasumsikan bahwa ada suatu model lokal linear antara citra penunjuk masukan I dengan keluaran citra yang sudah difilter q . Secara formal, model tersebut didefinisikan sebagai:

$$q_i = a_k I_i + b_k, \forall i \in \omega_k, \quad (11)$$

dengan q_i adalah piksel i dari citra keluaran, ω_k adalah *window* kotak dengan radius r yang berpusat di piksel k , I_i adalah piksel i dari citra penunjuk masukan, serta a_k dan b_k yang merupakan konstanta yang diasumsikan ada pada ω_k .

Untuk menentukan nilai dari konstanta a_k dan b_k , kita membutuhkan pembatas dari citra masukan yang ingin difilter p . Maka, kita akan memodelkan terlebih dahulu hubungan antara p dengan q dengan:

$$q_i = p_i - n_i \quad (12)$$

dengan n_i adalah n adalah *noise* yang ingin kita filter.

Kita ingin meminimalisir perbedaan nilai p dan q pada suatu piksel i dengan tetap mempertahankan mode linear di atas. Maka, kita dapat melakukannya dengan meminimalisir *cost function* berikut:

$$E(a_k, b_k) = \sum_{i \in \omega_k} ((a_k I_i + b_k - p_i)^2 + \epsilon a_k^2) \quad (13)$$

dengan $E(a_k, b_k)$ adalah *cost function* pada *window* ω_k dan ϵ adalah parameter regularisasi.

Kita dapat meminimalisir *cost function* di atas dengan menemukan nilai a_k dan b_k yang tepat. Kita dapat menemukan nilai-nilai tersebut dengan persamaan:

$$a_k = \frac{\frac{1}{|\omega|} \sum_{i \in \omega_k} I_i p_i - \mu_k \bar{p}_k}{\sigma_k^2 + \epsilon} \quad (14)$$

$$b_k = \bar{p}_k - a_k \mu_k \quad (15)$$

dengan μ_k and σ_k^2 adalah rata-rata dan variansi dari I pada *window* ω_k , $|\omega|$ adalah jumlah piksel pada *window* ω_k , dan \bar{p}_k adalah rata-rata dari piksel-piksel p pada *window* ω_k yang didefinisikan dengan:

$$\bar{p}_k = \frac{1}{|\omega|} \sum_{i \in \omega_k} p_i \quad (16)$$

Namun, sebuah piksel i pasti akan terlibat pada komputasi *window-window* yang mengandung i dan saling tumpang tindih pada gambar sehingga nilai q_i yang kita dapatkan dari persamaan (12) bisa berbeda-beda. Untuk mengatasinya, setelah kita berhasil menghitung nilai a_k dan b_k untuk semua *window* ω_k pada gambar, kita menghitung rata-rata dari semua kemungkinan nilai q_i dengan:

$$q_i = \frac{1}{|\omega|} \sum_{k|i \in \omega_k} a_k I_i + b_k \quad (17)$$

Karena $\sum_{k|i \in \omega_k} a_k = \sum_{i \in \omega_k}$ dan *window* berbentuk persegi yang simetris, maka persamaan di atas dapat kita ubah ke dalam bentuk:

$$q_i = \bar{a}_i I_i + \bar{b}_i \quad (18)$$

perhatikan bahwa \bar{a}_i dan \bar{b}_i adalah hasil dari *mean filtering* a_k dan b_k . Kita dapat memperhalus hasil estimasi transmisi yang sebelumnya didapat dengan persamaan (14), (15), dan (18).

Setelah kita berhasil mengetahui transmisi $t(x)$ yang sudah diperhalus, kita akan mencoba mengestimasi *radiance* J . Ketika $t(x)$ mendekati 0, maka nilai *Direct attenuation* $J(x)t(x)$ juga akan mendekati 0. Maka dari itu, kita memperkenalkan batas bawah t_0 untuk $t(x)$ agar nilai $t(x)$

tidak terlalu rendah. Batas bawah t_0 biasanya bernilai 0.1 Dengan menggunakan persamaan (1), kita bisa mendapatkan *radiance* J dengan persamaan:

$$\mathbf{J}(x) = \frac{\mathbf{I}(x) - \mathbf{A}}{\max(t(x), t_0)} + \mathbf{A} \quad (19)$$

Terakhir, kita akan mengestimasi cahaya atmosferik A . Langkah terakhir ini relatif lebih mudah ketimbang langkah-langkah sebelumnya. Pertama, kita mengambil 0.1% piksel dengan intensitas tertinggi pada *dark channel*. Dari piksel-piksel ini, kita ambil piksel dengan intensitas tertinggi pada citra masukan asli sebagai A .

Berikut adalah contoh proses DCP:



Gambar 2.4: Contoh proses DCP

Metode ini tentu saja bukan metode yang sempurna sehingga mempunyai keterbatasan. Kelemahan terbesar dari metode ini adalah ketika objek pada citra bebas *haze* mempunyai warna yang mirip atau sama dengan cahaya atmosfer dan tidak ada bayangan dari objek tersebut, metode ini akan mengestimasi transmisi objek tersebut terlalu rendah.

2.1.2 Metode *Deep Learning*: *Modified PDR-Net*

Deep learning yang kerap dielu-elukan akhir-akhir ini membuka beragam potensi yang mampu mengatasi kekurangan metode-metode konvensional pengolahan citra. Berkaitan dengan ini, *convolutional neural network*, atau lebih sering disebut dengan CNN, umum digunakan dalam jejaring pembelajaran mesin. CNN mampu mempelajari kemunculan-kemunculan pola dan fitur pada dataset gambar yang diberikan. Belakangan ini, muncul banyak implementasi CNN untuk melakukan *dehazing* atau penghilangan kabut pada citra, mengingat kabut adalah gangguan atmosferik yang umum hadir saat akuisisi citra.

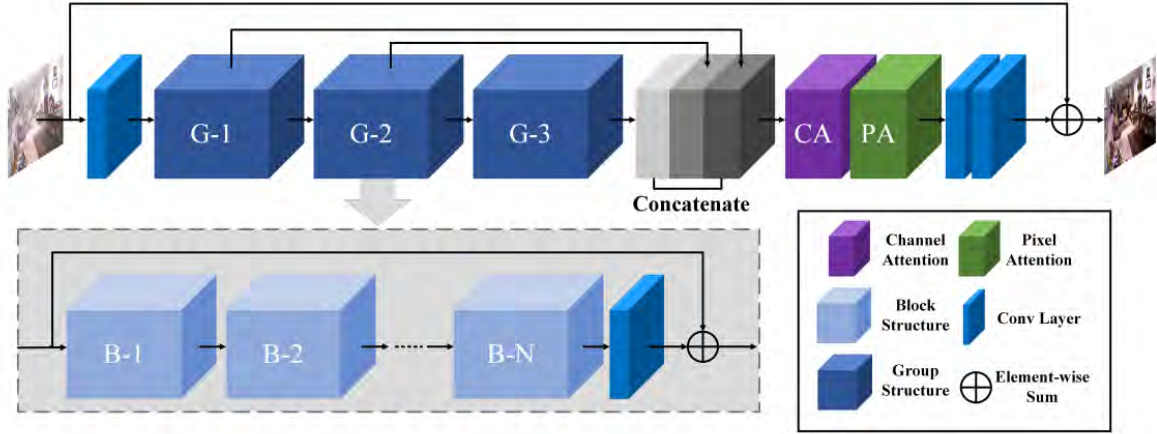
Modified PDR-Net (akan kami sebut dengan MPDR-Net) merupakan tajuk yang diusung oleh Hartanto dan Rahadiani (2021) untuk *neural network* mereka yang merupakan hasil modifikasi dari PDR-Net. PDR-Net sendiri adalah *neural network* yang dikembangkan oleh Li et al. (2020) dan merupakan kependekan dari *Perception-Inspired Single Image Dehazing Network with Refinement*. Fokus utama PDR-Net adalah menghasilkan citra bebas kabut yang lebih enak dilihat mata daripada metode-metode terdahulu dengan mengusung metode-metode perbaikan citra dan *perceptive loss*. Baik MPDR-Net maupun PDR-Net bekerja dengan dua *subnetwork*: *Haze-removal subnetwork* dan *refinement subnetwork*. Seperti namanya, *haze-removal subnetwork* akan menghasilkan prediksi citra tanpa kabut atau *haze* dan *refinement subnetwork* akan meningkatkan kualitas dari citra prediksi tersebut. Arsitektur inilah yang kemudian diadopsi oleh Hartanto dan Rahadiani dalam MPDR-Net.

Selain PDR-Net, FFA-Net usulan Qin (2019) juga ikut menginspirasi pembuatan MPDR-Net. FFA-Net, kependekan dari *Fusion Feature Attention Network* memanfaatkan sebuah komponen baru yakni mekanisme *Feature Attention* yang akan kita sebut FA. FA mengombinasikan mekanisme *Channel Attention* dan *Pixel Attention* yang mampu memberikan atensi berupa bobot yang berbeda untuk fitur-fitur berbeda kanal maupun piksel secara berurutan. Gambar berkabut cenderung memiliki distribusi kabut yang tak merata pada setiap piksel sehingga informasi yang bisa diambil dari tiap kanal maupun piksel tidak setara. FA mampu memperlakukan piksel yang berbeda secara berbeda sehingga bisa menerima informasi-informasi yang berbeda. Dengan ini, informasi yang dianggap penting bisa dipertahankan di lapisan-lapisan *network* yang lebih dalam.

Metode MPDR-Net yang diusulkan Hartanto dan Rahadiani memiliki beberapa modifikasi. Modifikasi-modifikasi tersebut di antaranya: Penggunaan *dilated convolution*, penggunaan *pyramid dilated convolution*, dan penambahan mekanisme atensi untuk kanal dan piksel. Per-

bandingan arsitektur antara MPDR-Net, PDR-Net, dan FFA-Net dilampirkan sebagaimana gambar-gambar di bawah disusun dengan uraian studi tentang bagaimana MPDR-Net bekerja. Perhatikan bahwa pada MPDR-Net, modul konvolusi utama memiliki notasi di atasnya yang merepresentasikan “ukuran kernel \times banyak feature map keluaran \times kecepatan dilasi pada *dilated convolution*”





Gambar 2.7: Arsitektur FFA-Net

1. Dilated Convolution

Penggunaan *dilated convolution* memperbesar *receptive field* secara multiplikatif dengan *cost* komputasi efisien. Ide utama *dilated convolution* adalah menyisipkan nilai kosong di antara piksel-piksel filter yang dipakai untuk konvolusi. Panjang jeda nilai kosong diatur oleh parameter yang disebut *dilation rate* atau kecepatan dilasi yang akan kita sebut d . Jeda yang dibuat di antara piksel filter akan senilai dengan $d - 1$. *Receptive field* suatu filter berperan cukup penting karena merepresentasikan seberapa besar informasi spasial yang dipakai dalam konvolusi.

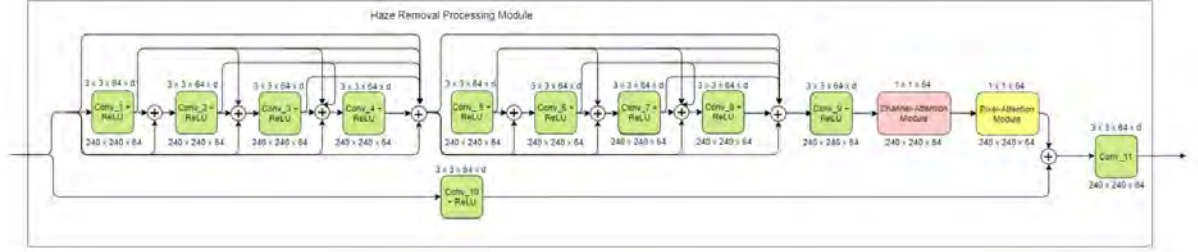
2. Pyramid Dilated Convolution

Penggunaan arsitektur ini dicirikan dengan adanya rentetan *dilated convolution* dengan nilai *dilation rate* yang meningkat secara bertahap. Hal ini dilakukan untuk mengatasi kehilangan terlalu banyak informasi spasial saat proses pembelajaran berlangsung yang mungkin terjadi ketika perbedaan *receptive field* terlalu jauh. Modul konvolusi dengan arsitektur piramidal seperti ini dipakai juga oleh Gridach dan Voiculescu (2020). Serupa dengan Gridach dan Voiculescu, Hartanto dan Rahadiani menggunakan operasi penjumlahan ketimbang konkatenasi pada akhir konvolusi. Operasi penjumlahan dinilai memiliki performa lebih baik dan mampu menangkap informasi dengan lebih baik pula.

3. Haze Removal and Refinement Processing Module

Modul konvolusi utama yang diusung dalam *network* (warna hijau dengan nama *Processing Module*) mengimplementasikan apa yang disebut dengan *skip connection*. *Skip connection* diaplikasikan dalam *Residual Neural Network* atau ResNet dan bekerja untuk mempertahankan informasi ke lapisan *network* yang lebih dalam dengan "melongkapi" lapisan-lapisan konvolusi. Dengan adanya hal ini, informasi mengenai piksel-piksel dengan *haze* yang cenderung tipis mampu dipertahankan. Arsitektur modul ini dapat dilihat pada gambar di bawah.

Dalam modul ini, ada lapisan konvolusi yang menggunakan *activation function* ReLU dan ada pula yang tidak. Lapisan konvolusi dengan ReLU memiliki fungsi meningkatkan mutu *feature map* yang dihasilkan oleh lapisan konvolusi itu sendiri sebelum kemudian diproses

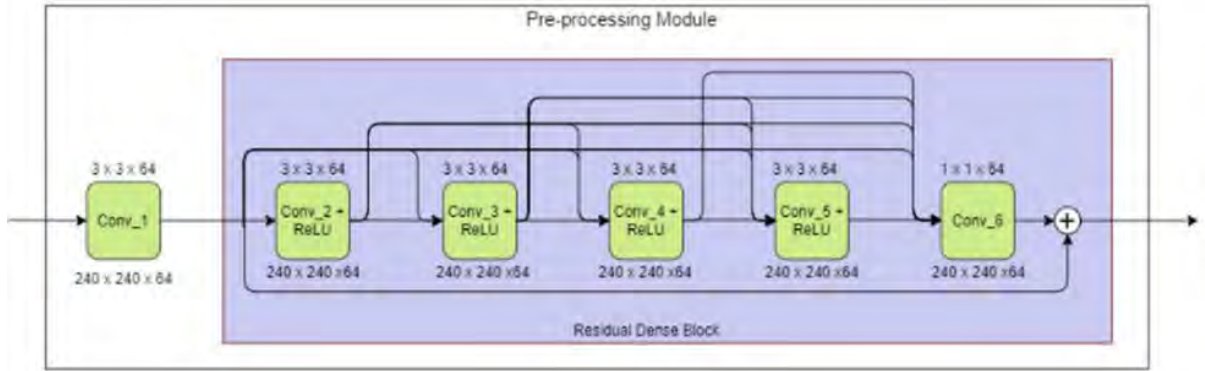


Gambar 2.8: Arsitektur *Processing Module*

oleh mekanisme atensi. Lapisan konvolusi yang tidak menggunakan ReLU berfungsi untuk *latent reconstruction* yang dapat dilakukan dengan melakukan penjumlahan pada hasil rentetan konvolusi sebelumnya dengan hasil *skip connection*.

4. *Pre and Post-Processing Module*

Modul-modul ini (ditandai dengan nama sama pada gambar) diimplementasikan dengan RDB atau *Residual Dense Block* dan diadopsi dari GridDehaze-Net. Modul ini akan menghasilkan 64 *feature map* dan bertujuan untuk meningkatkan relevansi hasil dan efisiensi dari tahap *pre-processing* maupun *post-processing*. Modul *pre-processing* diimplementasikan dengan sebuah lapisan konvolusi tanpa ReLU dan diikuti RDB berisikan lapisan-lapisan konvolusi dengan ReLU. Sebaliknya, modul *post-processing* diimplementasikan dengan RDB yang sama dan diikuti oleh sebuah lapisan konvolusi tanpa ReLU. Arsitektur umum modul *pre-processing* dapat dilihat sebagaimana berikut.



Gambar 2.9: Arsitektur *Pre-Processing Module*

5. Mekanisme Atensi: *Channel Attention Module*

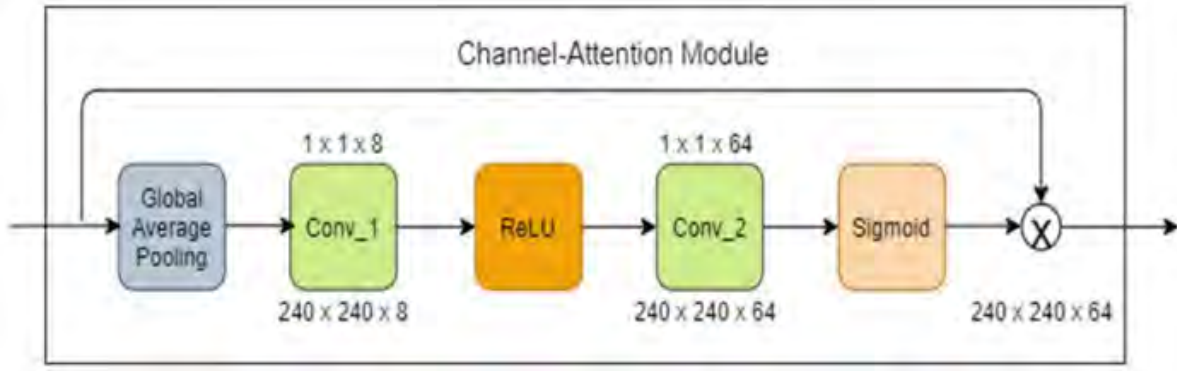
Kedua modul mekanisme atensi, sebagaimana dijelaskan pada beberapa paragraf sebelumnya, diadopsi dari FFA-Net. Akan dibahas mekanisme atensi untuk kanal terlebih dahulu.

Modul ini (ditandai dengan modul merah) akan memberikan bobot tersendiri untuk masing-masing kanal sehingga memberikan informasi yang lebih mendalam tentang hubungan kanal warna dan gambar berkabut. Pada awalnya, informasi diproses dengan menggunakan *Global Average Pooling*. *Global Average Pooling* beroperasi dengan mengkalkulasikan rerata keluaran dari *feature map* masukan pada lapisan sebelumnya dan

menghasilkan informasi dengan ukuran yang lebih ringkas. *Global Average Pooling* direpresentasikan secara matematis sebagai berikut.

$$g_c = H_p(F_c) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W X_c(i, j) \quad (20)$$

Global Average Pooling H_p akan menghasilkan keluaran berbentuk $C \times 1 \times 1$ dengan C merupakan banyak kanal masukan dari masukan F_c . $X_c(i, j)$ menandakan nilai pada kanal C pada koordinat piksel (i, j) . Hasil ini kemudian diproses dengan sebuah dua lapisan konvolusi, yang pertama diikuti langsung oleh *activation function* ReLU dan yang kedua oleh fungsi sigmoid untuk menghasilkan bobot-bobot yang berbeda. Informasi yang diperoleh dari modul atensi kanal ini bersifat *channel-wise*, global, dan spasial. Arsitektur modul ini digambarkan oleh diagram berikut.



Gambar 2.10: Arsitektur *Channel Attention Module*

Perhitungan selanjutnya dapat direpresentasikan sebagai persamaan-persamaan berikut.

$$CA_c = \sigma(\text{Conv}(\delta(\text{Conv}(g_c)))) \quad (21)$$

$$\text{output} = CA_c \otimes F_c \quad (22)$$

Dengan σ adalah fungsi sigmoid dan δ adalah fungsi ReLU. Bobot yang didapat kemudian dikalikan secara *element-wise* dengan masukan awal F_c . Hasil akhir akan berbentuk 64 *feature map*.

6. Mekanisme Atensi: *Pixel Attention Module*

Modul ini (ditandai dengan modul kuning) bertujuan untuk menangkap informasi dari piksel dengan kabut tinggi dengan lebih baik. Arsitektur modul ini serupa dengan modul *Channel Attention* hanya saja tidak dilakukan *Global Average Pooling* pada tahap awalnya. Hasil sebelum dilakukan operasi *element-wise* akan berbentuk $1 \times H \times W$ dengan H dan W merupakan dimensi dari gambar masukan dan dapat dituliskan dalam notasi matematis sebagaimana berikut, dengan I sebuah masukan.

$$PA = \sigma(\text{Conv}(\delta(\text{Conv}(I)))) \quad (23)$$

Pemrosesan lebih lanjut dari modul ini dapat direpresentasikan sebagai persamaan-persamaan berikut, dengan PA bobot yang didapatkan dan I masukan awal.

$$output = F \otimes PA \quad (24)$$

7. Loss Function

Pada dasarnya, *neural network* belajar dengan meminimalkan galat yang dihasilkan oleh *loss function*. Pemilihan *loss function* yang tepat dapat menghasilkan performa yang optimum. *Loss function* yang dibahas di sini adalah L1 yang dapat direpresentasikan sebagaimana berikut.

$$L_1 = \frac{1}{N} \sum_{x=1}^N \sum_{i=1}^3 F_s(I_i(x) - G_i(x)) \quad (25)$$

dengan

$$F_s(e) = \begin{cases} 0.5e^2, & \text{if } |e| < 1 \\ |e| - 0.5, & \text{otherwise} \end{cases} \quad (26)$$

N merepresentasikan keseluruhan piksel gambar. $I_i(x)$ merepresentasikan nilai intensitas pada piksel x dari gambar hasil prediksi dan $G_i(x)$ dari gambar *ground truth*. Selain menggunakan L1, MPDR-Net juga menggunakan *perceptual loss*. Berdasarkan rujukan utama MPDR-Net, *perceptual loss* bertujuan untuk meminimalkan perbedaan antara gambar hasil prediksi dan *ground truth* secara persepsi, memperkuat *feature* halus, dan mempertahankan informasi akan warna. Untuk menghitung *perceptual loss*, MPDR-Net menggunakan bantuan model VGG16 dengan mengekstraksi tiga lapisan aktivasinya dari model tersebut. *Perceptual loss* dapat dituliskan sebagaimana berikut.

$$L_p = \sum_{j=1}^3 \frac{1}{C_j H_j W_j} \|\phi_j(I) - \phi_j(G)\| \quad (27)$$

$C_j H_j W_j$ menandakan *featuremap* dari gambar hasil prediksi dan *ground truth* dan ϕ_j merepresentasikan *feature* persepsi VGG16. Kedua *loss function* kemudian dikombinasikan sebagaimana berikut.

$$L_{total} = L_1 + \lambda L_p \quad (28)$$

Parameter λ digunakan untuk mengatur bobot antara *perceptual loss* dan L1. Pada MPDR-Net, dipilih $\lambda = 0.04$

2.2 Desain Eksperimen

Tahapan dari penelitian kami adalah sebagai berikut:

1. Mempersiapkan implementasi DCP dan MPDR-Net
2. Mempersiapkan *test cases*
3. Memasukkan *test cases* ke dalam algoritma DCP dan MPDR-Net
4. Mencatat waktu kedua metode pada setiap *test case*
5. Mencatat skor SSIM untuk setiap hasil dari komputasi pada *test case*
6. Melakukan analisis pada hasil percobaan

Untuk implementasi DCP, kami menggunakan implementasi yang sudah dibuat oleh He Zhang (2021). Sedangkan untuk MPDR-Net, kami mendapatkan implementasi metode tersebut dari Hartanto dan Rahadiani (2021). Tautan *repository* dari kedua implementasi tersebut bisa dilihat pada bagian referensi di akhir laporan ini.

Kami akan menggunakan *test case* berupa 45 pasang citra yang merupakan pasangan gambar dengan haze beserta gambar yang sama tanpa haze. Citra-citra tersebut kami dapatkan dari dataset O-haze yang tersedia untuk umum. Selain menggunakan citra dari O-haze, kami juga mengambil tiga citra berkabut yang diambil pada situasi nyata sebagai masukan tambahan bagi kedua metode tersebut. Setiap *test case* tersebut akan kami jadikan masukan bagi algoritma DCP. Untuk MPDR-Net, kami memilih secara acak citra-citra pada O-haze untuk dijadikan *training set* *validation set*, dan *testing set* dengan persentase masing-masing 64% (29 citra), 16% (7 citra) dan 20% (9 citra).



















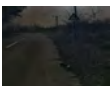







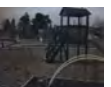









Setiap percobaan tersebut akan dicatat waktunya dengan memanfaatkan modul Time pada bahasa pemrograman Python. Hasil dari percobaan pada kedua metode tersebut akan kami analisis dan bandingkan satu sama lain. Terakhir, kami akan menarik kesimpulan berdasarkan analisis dan perbandingan yang telah dilakukan sebelumnya.

Untuk metode pengukuran kualitas hasil, kami memilih menggunakan *structural similarity index measure* (SSIM). Menurut kami, SSIM sangat baik digunakan pada percobaan ini karena SSIM dapat mengukur kesamaan dari dua citra sampai pada detail kesamaan struktur, pencahayaan, dan kontras. Karena hal tersebut SSIM lebih dapat menyimulasikan persepsi manusia terhadap dua gambar ketimbang metode lain seperti *mean squared error* (MSE) dan *peak signal-to-noise ratio* (PSNR). Keunggulan SSIM ketimbang metode lain juga didukung oleh penelitian Wang et al. (2004) yang merupakan pencipta dari metode SSIM.

2.3 Hasil Eksperimen

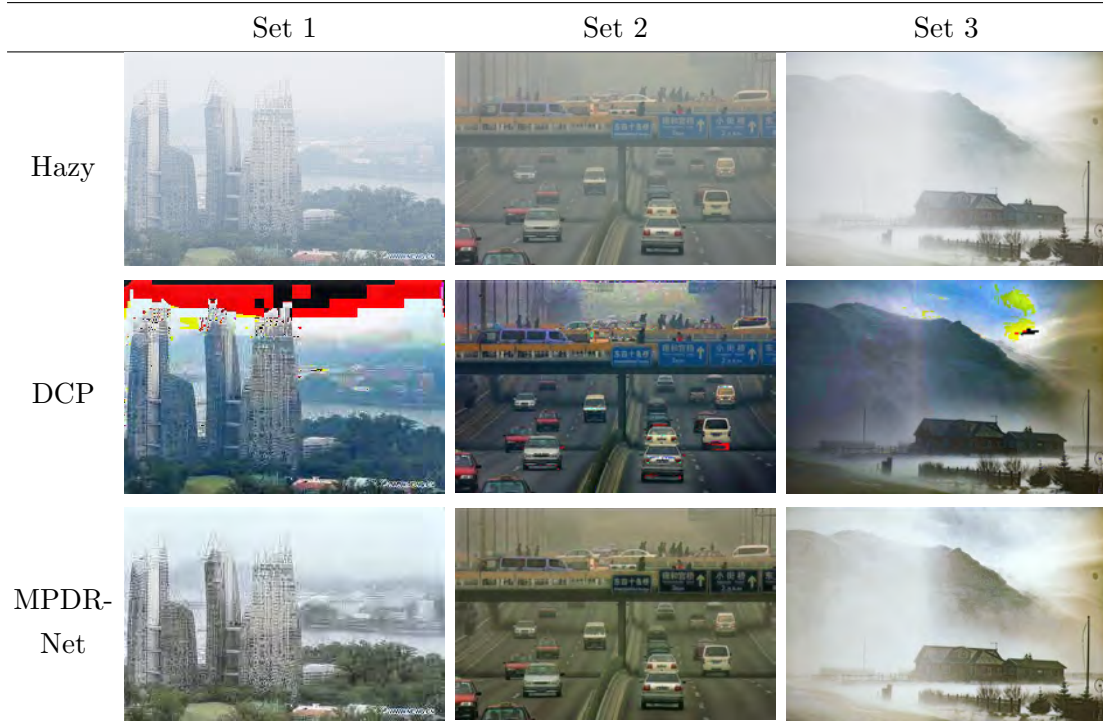
Tabel di bawah menunjukkan hasil perbandingan penggunaan DCP dengan MPDR-Net. Dapat dilihat pada set 4, 14, 41, dan 45 citra yang dihasilkan MPDR-Net memiliki warna yang lebih gelap dibandingkan *ground truth* citranya, seolah-olah warna biru pada citra tersebut hilang. Sedangkan pada set 8 dan 20 masih terdapat sedikit kabut pada citra, tetapi objek masih terlihat cukup jelas. Pada set 21, 22, dan 23, citra yang dihasilkan sudah cukup mirip dengan *ground truth* citra tersebut. Secara keseluruhan, citra yang dihasilkan MPDR-Net sudah menyerupai citra *ground truth* meskipun belum sempurna.

Sedangkan hampir seluruh citra yang dihasilkan DCP membuat kabut menjadi agak kebiruan. Terdapat suatu keanehan pada set 14 di mana langit pada citra berubah menjadi warna hijau. Hal tersebut menyebabkan citra yang dihasilkan DCP masih belum menyerupai citra *ground truth*.

	Set 4	Set 8	Set 14	Set 20	Set 21	Set 22	Set 23	Set 41	Set 45
Hazy									
DCP									
MPDR-Net									
GT									

Tabel 2.1: Perbandingan Restorasi Citra antara DCP dan MPDR-Net pada Dataset Test.

Kami juga melakukan percobaan terhadap kedua metode dengan memasukkan gambar dari Google. Kami belum sempat melakukan Psycho-Visual Testing, namun dapat dilihat bahwa citra yang dihasilkan MPDR-Net secara umum lebih baik dibandingkan dengan DCP karena citra yang dihasilkan tidak memiliki piksel-piksel rusak.



Tabel 2.2: Perbandingan Restorasi Citra antara DCP dan MPDR-Net pada Citra Kondisi Nyata.

Berikut adalah tabel yang menunjukkan SSIM dan *runtime* dari setiap set yang dijalankan DCP dan MPDR-Net.

Set	DCP		MPDR-Net	
	SSIM	Runtime (s)	SSIM	Runtime (s)
4	0.14	2.40	0.66	1.57
8	0.25	2.21	0.81	1.52
14	0.29	4.06	0.86	1.67
20	0.29	3.40	0.82	1.53
21	0.16	4.86	0.88	1.78
22	0.34	4.91	0.95	1.74
23	0.59	2.66	0.93	1.36
41	0.33	1.69	0.94	1.17
45	0.21	2.31	0.83	1.27
Rata-rata	0.29	3.17	0.85	1.51

Tabel 2.3: Perbandingan SSIM dan *runtime* DCP dan MPDR-Net pada *testing set* MPDR-Net

Dapat dilihat dari tabel di atas bahwa berdasarkan pengukuran skor SSIM, MPDR-Net secara konsisten jauh mengungguli DCP dalam melakukan *single image dehazing*. Selain itu, rata-rata kecepatan runtime MPDR-Net dalam melakukan restorasi citra juga lebih cepat dari kecepatan runtime DCP.

Namun, untuk melakukan pelatihan terhadap model MPDR-Net membutuhkan waktu yang cukup lama. Pelatihan model MPRD-Net dengan 36 citra *training set* dari O-haze yang kami lakukan pada eksperimen ini membutuhkan waktu sekitar 234 menit. Hal ini dikarenakan pelatihan model MPDR-Net yang menggunakan *deep learning* lebih kompleks dan memakan sumber daya yang jauh lebih besar untuk melatih modelnya. Sedangkan, metode DCP tidak memerlukan pelatihan model sehingga kecepatan *runtime* dari DCP jauh lebih cepat ketimbang kecepatan *runtime* MPDR-Net pada tahap pelatihan model.

Bab 3

Penutup

3.1 Kesimpulan

Kesimpulan kami adalah bila kita memiliki sumber daya yang besar, waktu, dan data yang cukup untuk melakukan pelatihan model, maka MPDR-Net sangat cocok digunakan untuk melakukan *single image dehazing* karena akurasinya yang relatif sangat tinggi. Namun, bila kita tidak mempunyai sumber daya dan data yang cukup, maka kita dapat menggunakan DCP untuk mengambil aproksimasi citra bebas *haze* dari sebuah citra yang memiliki *haze*.

3.2 Saran

Penelitian ini diharapkan dapat memberikan wawasan akan beberapa metode *image dehazing* yang mencakup teknik konvensional maupun *deep learning*. Kemudian diharapkan dapat dikembangkan sebuah metode *image dehazing* yang memberikan hasil yang lebih akurat di kemudian hari dari penelitian ini.

3.3 Refleksi Kelompok

Selama melakukan penelitian, terdapat beberapa kendala yang sempat menghalangi kelancaran proses penelitian. Kendala utama adalah keterbatasan sumber daya teknologi untuk menjalankan program implementasi MPDR-Net mengingat komputasinya membutuhkan spesifikasi mesin yang cukup tinggi agar dapat berjalan dengan waktu singkat.

Kendala lainnya yang cukup memberatkan adalah waktu pengerjaan penelitian ini karena

berjalan bersamaan dengan beragam tugas kelompok dan tugas akhir dari mata kuliah lain. Karena hal ini, peneliti menyadari bahwa seharusnya peneliti bisa lebih baik dalam mengatur waktu dengan memulai penelitian lebih awal sehingga beban penelitian tersebar lebih merata dalam jangka waktu pengerjaan.

Bibliografi

- [Brownlee, 2019] Brownlee, J. (2019). Loss and loss functions for training deep learning neural networks.
- [Gridach and Voiculescu, 2020] Gridach, M. and Voiculescu, I. (2020). Oxendonet: A dilated convolutional neural networks for endoscopic artefact segmentation.
- [Hartanto and Rahadiani, 2021] Hartanto, C. A. and Rahadiani, L. (2021). Single image dehazing using deep learning. *JOIV : International Journal on Informatics Visualization*, 5(1):76.
- [He et al., 2010] He, K., Sun, J., and Tang, X. (2010). Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2341–2353.
- [He et al., 2013] He, K., Sun, J., and Tang, X. (2013). Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409.
- [Lee et al., 2016] Lee, S., Yun, S., Nam, J.-H., Won, C. S., and Jung, S.-W. (2016). A review on dark channel prior based image dehazing algorithms. *EURASIP Journal on Image and Video Processing*, 2016(1).
- [Li et al., 2019] Li, C., Chunle, G., Guo, J., Han, P., Fu, H., and Cong, R. (2019). Pdr-net: Perception-inspired single image dehazing network with refinement. *IEEE Transactions on Multimedia*, PP:1–1.
- [Li et al., 2020] Li, C., Guo, C., Guo, J., Han, P., Fu, H., and Cong, R. (2020). Pdr-net: Perception-inspired single image dehazing network with refinement. *IEEE Transactions on Multimedia*, 22(3):704–716.
- [Liu et al., 2019] Liu, X., Ma, Y., Shi, Z., and Chen, J. (2019). Griddehazenet: Attention-based multi-scale network for image dehazing.
- [n.a., 2019] n.a. (2019). An introduction to global average pooling in convolutional neural networks.
- [Pröve, 2018] Pröve, P.-L. (2018). An introduction to different types of convolutions in deep learning.

- [Qin et al., 2019] Qin, X., Wang, Z., Bai, Y., Xie, X., and Jia, H. (2019). Ffa-net: Feature fusion attention network for single image dehazing.
- [Wang et al., 2004] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.