

LAPORAN FINAL PROJECT PCD

DIGIPRO

APLIKASI KOMPILASI FILTER UNTUK EASY ACCESS

IMAGE FILTERING



Disusun Oleh :

Alif Adrian Anzary - 5025201274

Mohammad Fadhil Rasyidin Parinduri - 5025201131

Bimantara Tito Wahyudi - 5025201227

TEKNIK INFORMATIKA

FAKULTAS TEKNIK ELEKTRO DAN INFORMATIKA CERDAS

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

BAB I

PENDAHULUAN

1. Latar Belakang

Pengolahan citra digital adalah cabang ilmu komputer yang berkaitan dengan pemrosesan, analisis, dan manipulasi gambar atau citra yang direpresentasikan dalam bentuk digital. Mata kuliah Pengolahan Citra Digital merupakan bagian penting dalam bidang komputer visi dan memiliki banyak aplikasi praktis di berbagai industri, seperti pengenalan wajah, pengenalan pola, analisis medis, pengenalan tulisan tangan, dan banyak lagi.

Fotografi dan pengeditan gambar telah menjadi bagian penting dalam kehidupan sehari-hari, terutama dengan kemajuan teknologi yang semakin canggih. Pengguna sering mencari aplikasi yang dapat memberikan efek kreatif pada foto-foto mereka. Namun, saat ini terdapat banyak aplikasi yang menawarkan efek-efek tersebut secara terpisah. Oleh karena itu, terdapat kebutuhan untuk mengembangkan sebuah aplikasi yang menggabungkan berbagai efek secara menyeluruh.

2. Tujuan

Tujuan dari proposal ini adalah mengembangkan sebuah aplikasi yang menawarkan berbagai efek dan filter dalam satu aplikasi yang mudah digunakan. Aplikasi ini akan memberikan pengguna kemampuan untuk mengunggah foto dan langsung menerapkan efek-efek kreatif yang diinginkan. Aplikasi ini akan memberikan pengguna kebebasan kreatif untuk menghasilkan foto yang menarik dan unik tanpa perlu beralih antara beberapa aplikasi yang berbeda.

3. Manfaat

- a. Kemudahan Penggunaan: Aplikasi pengolahan gambar dengan berbagai efek yang terintegrasi akan menyediakan antarmuka yang sederhana dan mudah digunakan. Pengguna akan dapat mengakses berbagai efek hanya dengan beberapa kali klik, tanpa harus mempelajari atau beralih antara beberapa aplikasi terpisah.

- b. Kreativitas yang Ditingkatkan: Dengan adanya berbagai efek dan filter dalam satu aplikasi, pengguna akan memiliki lebih banyak pilihan untuk menciptakan hasil foto yang kreatif. Mereka dapat menggabungkan efek-efek yang berbeda dan melihat langsung hasilnya.
- c. Efisiensi Waktu: Dengan menggabungkan berbagai efek dalam satu aplikasi, pengguna akan menghemat waktu yang sebelumnya digunakan untuk mencari, mengunduh, dan membuka aplikasi-aplikasi terpisah. Mereka dapat langsung memproses foto mereka dengan berbagai efek yang diinginkan hanya dalam satu aplikasi.

BAB II

TINJAUAN PUSTAKA

1. Aplikasi Pengolahan Gambar dan Pengeditan Gambar

Aplikasi Pengolahan Gambar telah berkembang pesat dalam beberapa tahun terakhir. Mereka memungkinkan pengguna untuk mengambil foto langsung dari perangkat mereka dan memberikan kemampuan untuk mengedit dan memperbaiki gambar dengan berbagai efek dan filter. Tinjauan ini akan mengulas beberapa pengeditan gambar yang populer dan fitur-fitur yang mereka tawarkan.

1. Instagram: Instagram adalah salah satu aplikasi sosial media yang sangat populer untuk berbagi foto. Aplikasi ini menawarkan berbagai efek filter yang dapat diaplikasikan pada foto sebelum dibagikan. Instagram juga memiliki fitur pengeditan gambar dasar seperti koreksi warna, kecerahan, dan ketajaman.
2. Adobe Lightroom: Lightroom adalah salah satu aplikasi pengeditan gambar yang populer di kalangan fotografer. Aplikasi ini menawarkan kontrol yang lebih detail terhadap koreksi warna, kurva tonal, dan pengaturan lainnya. Lightroom juga menyediakan kemampuan untuk menyimpan preset pengeditan yang dapat diterapkan pada berbagai foto.

2. Algoritma dan Metode Pengolahan Gambar

Algoritma dan metode pengolahan gambar merupakan komponen kunci dalam pengembangan aplikasi pengolahan gambar dengan berbagai efek. Di bawah ini adalah beberapa algoritma dan metode yang relevan yang dapat digunakan untuk mengimplementasikan efek-efek yang diinginkan dalam aplikasi pengolahan gambar:

1. Filtering: Filtering adalah teknik dasar dalam pengolahan gambar yang melibatkan operasi pada piksel-piksel gambar. Filter linear seperti filter rata-rata (average filter), filter Gaussian, dan filter sharpening dapat digunakan untuk mencapai efek seperti blur, pengurangan noise, dan peningkatan ketajaman.
2. Transformasi Warna: Transformasi warna memungkinkan manipulasi warna dalam gambar. Salah satu contoh transformasi warna adalah konversi gambar menjadi skala abu-abu (grayscale) dengan mengambil rata-rata komponen warna merah, hijau, dan

biru dari setiap piksel. Transformasi warna juga dapat mencakup penyesuaian kontras, kecerahan, dan saturasi.

3. Segmentasi Gambar: Segmentasi gambar membagi gambar menjadi beberapa bagian berdasarkan atribut tertentu seperti warna, tekstur, atau intensitas. Algoritma segmentasi seperti metode Otsu atau metode pemisahan berdasarkan ambang (thresholding) dapat digunakan untuk memisahkan objek dari latar belakang atau untuk mengidentifikasi fitur tertentu dalam gambar.

3. Framework dan Library Pengolahan Gambar

Dalam pengembangan aplikasi pengolahan gambar dengan berbagai efek, penggunaan framework dan library pengolahan gambar dapat sangat membantu untuk mempercepat proses pengembangan dan mengimplementasikan fitur-fitur yang kompleks. Berikut ini adalah beberapa framework dan library yang dapat digunakan dalam pengembangan aplikasi pengolahan gambar:

1. OpenCV: OpenCV (Open Source Computer Vision Library) adalah salah satu library pengolahan gambar yang paling populer dan banyak digunakan. OpenCV menyediakan berbagai fungsi dan algoritma pengolahan gambar yang kuat, termasuk manipulasi gambar, filtering, segmentasi, deteksi objek, dan banyak lagi. OpenCV mendukung berbagai bahasa pemrograman seperti C++, Python, dan Java, sehingga dapat digunakan dalam pengembangan aplikasi pengolahan gambar lintas platform.
2. TensorFlow: TensorFlow adalah framework machine learning yang sangat populer, tetapi juga dapat digunakan untuk pengolahan gambar. Framework ini menyediakan berbagai alat dan fungsi yang kuat untuk membangun dan melatih model deep learning dalam pengolahan gambar. TensorFlow memiliki dukungan untuk arsitektur jaringan saraf seperti Convolutional Neural Networks (CNN) dan menyediakan pre-trained model yang dapat digunakan untuk tugas-tugas seperti deteksi objek, segmentasi, dan stylization.
3. PyTorch: PyTorch adalah framework deep learning yang semakin populer dalam pengolahan gambar. PyTorch menyediakan kemudahan dalam mengembangkan dan melatih model deep learning dengan dukungan penuh untuk GPU. Framework ini memiliki banyak alat dan fungsi untuk pengolahan gambar, termasuk transformasi gambar, jaringan saraf konvolusi, dan visualisasi hasil.

BAB III

METODOLOGI

1. Pemilihan Filter

Dalam pemilihan filter, langkah-langkah yang perlu dilakukan adalah mengidentifikasi kebutuhan pengguna berdasarkan penelitian dan tren terbaru dalam fotografi, meninjau literatur untuk mempelajari jenis filter yang tersedia, menganalisis efek yang akan diimplementasikan, dan memilih filter yang paling relevan berdasarkan kompleksitas, kinerja, dan kepuasan pengguna.

2. Pengembangan Aplikasi

Dalam pengembangan aplikasi, langkah-langkah yang perlu dilakukan meliputi desain antarmuka pengguna yang intuitif, implementasi fitur pengunggahan gambar, integrasi filter yang telah dipilih menggunakan algoritma pengolahan gambar yang relevan, pengembangan fitur pengeditan gambar, dan pengujian fungsionalitas untuk memastikan semua fitur berfungsi dengan baik.

3. Evaluasi Hasil

Dalam evaluasi hasil, langkah-langkah yang perlu dilakukan adalah melakukan uji kualitas gambar untuk mengevaluasi hasil gambar yang dihasilkan oleh aplikasi menggunakan berbagai filter dan efek, serta membandingkannya dengan gambar asli. Selain itu, dapat dilakukan uji kepuasan pengguna melalui wawancara atau survei untuk mendapatkan umpan balik pengguna terkait pengalaman menggunakan aplikasi dan kualitas hasil gambar, dengan tujuan mengidentifikasi area perbaikan yang mungkin diperlukan.

BAB IV

HASIL DAN PEMBAHASAN

1. Pemilihan Filter

Filter yang akan digunakan dalam aplikasi, adalah sebagai berikut:

a. Adjustment

- **Crop**

Memilih bagian tertentu dari suatu gambar dan menghapus sisanya, memangkas gambar secara efektif.

- **Rotate**

Memutar gambar searah jarum jam atau berlawanan arah jarum jam dengan sudut tertentu.

- **Flip**

Membalik gambar secara horizontal atau vertikal, membalik urutan piksel di sepanjang sumbu yang dipilih.

- **Brigthness**

Menyesuaikan kecerahan atau kegelapan gambar secara keseluruhan, menjadikannya lebih terang atau lebih gelap.

- **Contrast**

Mengubah perbedaan antara area terang dan gelap pada gambar, meningkatkan perbedaan di antara keduanya.

- **Sharpness**

Meningkatkan tingkat detail dan kerenyahan pada gambar dengan menekankan tepi dan komponen frekuensi tinggi.

b. Filter

- **Blur**

Filter blur digunakan untuk mengurangi tingkat detail pada gambar, sehingga menghasilkan tampilan yang lebih halus. Ini bekerja dengan meratakan nilai warna dari piksel tetangga, secara efektif mengurangi tepi tajam dan detail halus.

- **Invert**

Filter invert digunakan untuk membalikkan warna gambar. Ini mengubah piksel terang menjadi piksel gelap dan sebaliknya, pada dasarnya menciptakan versi negatif dari gambar aslinya.

- Dual Channel

Filter dual channel digunakan untuk memanipulasi gambar dengan memisahkannya menjadi dua channel. Setiap channel mewakili aspek gambar yang berbeda, dan dengan menyesuaikan properti saluran ini secara terpisah, dapat diciptakan berbagai efek atau menyempurnakan elemen gambar tertentu.

- Edge Enhance

Filter edge enhance dirancang untuk menekankan tepi dan batas dalam gambar. Ini bekerja dengan mendeteksi perubahan intensitas piksel dan meningkatkan kontras di sepanjang tepi ini, menjadikannya lebih menonjol dan jelas.

- Emboss

Filter emboss digunakan untuk membuat tampilan tiga dimensi dengan menyorot tepi dan kontur objek dalam gambar. Ini mencapai efek ini dengan memetakan perbedaan intensitas antara piksel tetangga ke skala abu-abu, memberikan ilusi area yang ditinggikan atau diukir.

- Contour

Filter kontur mirip dengan filter penyempurnaan tepi, karena ini bertujuan untuk menyempurnakan tepi pada gambar. Namun, alih-alih sekadar meningkatkan kontras, filter kontur menerapkan efek gaya yang menekankan kontur objek dengan garis tepi atau siluet yang jelas.

c. Advanced

- Moiré Restore

Pola Moiré adalah pola interferensi bergelombang yang tidak diinginkan yang dapat terjadi saat memindai atau memotret gambar dengan detail berulang. Moiré restore adalah fitur yang membantu mengurangi atau menghilangkan pola-pola ini dan mengembalikan kejernihan dan detail gambar aslinya. Pendekatannya adalah dengan menggunakan notch reject filter. Akan tetapi filter notch reject membutuhkan bantuan pengguna untuk memilih lokasi

yang akan difilter pada domain frekuensi. Untuk mengatasi ketidaknyamanan tersebut, dapat digunakan algoritma yang mendeteksi adanya puncak frekuensi pada spektrum yang bukan merupakan frekuensi utama dan kemudian menerapkan notch reject filter pada lokasi tersebut.

- Face Detection

Deteksi wajah adalah teknologi yang mengidentifikasi dan menemukan wajah manusia dalam gambar. Digunakan model machine learning yang dilatih dengan dataset yang telah tersedia untuk menghasilkan model yang dapat mendeteksi adanya wajah manusia dalam gambar.

- Document Scan

Document scan mengacu pada proses modifikasi citra sedemikian rupa hingga terlihat seperti gambar yang diambil menggunakan scanner. Document scan sering melibatkan pemotongan otomatis, penyempurnaan gambar, dan pengenalan karakter optik (OCR) untuk meningkatkan keterbacaan dan mengekstrak teks untuk tujuan pemrosesan atau penyimpanan lebih lanjut.

2. Pengembangan Aplikasi

Antarmuka aplikasi ini dikembangkan dari awal dengan library PyGame untuk antarmuka dan Tkinter untuk open dan save dialog serta tanpa menggunakan framework, source code orang lain, dan library bantuan lainnya. Dari 15 fitur filter yang sebelumnya telah dipilih, hanya terdapat 10 filter yang dikembangkan dengan 9 diantaranya berhasil diimplementasikan. Berikut di antaranya:

- a. Adjustment

- Brightness

Diimplementasikan menggunakan tanpa library selain PyGame. Seluruh pixel ditambah dengan nilai yang ditentukan oleh amount. Dilakukan pengecekan agar filter hasil tidak keluar dari constraint.

```
def brighten(self, amount=10):
    image_array = pygame.surfarray.array3d(self.previewImage)
    image_array = image_array * (1.0 + (amount / 20 * 0.8))
    if amount > 0:
        image_array[image_array > 255] = 255
```

```
elif amount < 0:
    image_array[image_array < 0] = 0
self.previewImage = pygame.surfarray.make_surface(image_array)
```

- Contrast

Diimplementasikan menggunakan bantuan library Pillow. Citra difilter menggunakan fungsi enhance dari module ImageEnhance.Contrast dengan amount sebagai intensitas filter yang digunakan.

```
def contrast(self, amount=10):
    image_array = pygame.surfarray.array3d(self.previewImage)
    image = Img.fromarray(image_array)
    enhancer = ImageEnhance.Contrast(image)
    image = enhancer.enhance(1 + (amount / 20))
    self.previewImage = pygame.surfarray.make_surface(np.array(image))
```

- Sharpness

Diimplementasikan menggunakan bantuan library Pillow. Citra difilter menggunakan fungsi enhance dari module ImageEnhance.Sharpness dengan amount sebagai intensitas filter yang digunakan.

```
def sharp(self, amount=10):
    image_array = pygame.surfarray.array3d(self.previewImage)
    image = Img.fromarray(image_array)
    enhancer = ImageEnhance.Sharpness(image)
    image = enhancer.enhance(1 + (amount / 20))
    self.previewImage = pygame.surfarray.make_surface(np.array(image))
```

b. Filter

- Blur

Diimplementasikan melalui domain frekuensi. Filter yang digunakan merupakan filter low pass dengan fungsi gaussian untuk hasil yang lebih mulus.

```
def blur(self, D0=10):
    F_domain = []
    M_shape, N_shape, L_shape = self.previewSpectrum.shape
    for i in range(3):
        H = np.zeros((M_shape, N_shape), dtype=np.float32)
```

```

        F_shift = self.previewSpectrum[:, :, i]
        for u in range(M_shape):
            for v in range(N_shape):
                D = np.sqrt((u - M_shape / 2) ** 2 + (v - N_shape / 2) ** 2)
                H[u, v] = math.e ** (-1 * D ** 2 / (2 * D0 ** 2))
    G_shift = F_shift * H
    F_domain.append(G_shift)
    self.previewSpectrum = np.dstack([F_domain[0].astype('complex_'),
    F_domain[1].astype('complex_'), F_domain[2].astype('complex_')])

```

Hasil spectrum pada self.previewSpectrum akan di-convert menjadi image menggunakan fungsi self.updateImage.

- Invert

Diimplementasikan menggunakan tanpa library selain PyGame. Seluruh pixel diganti menjadi $255 - \text{pixel}$.

```

def invert(self):
    print("Executed Invert")
    image_array = pygame.surfarray.array3d(self.previewImage)
    H = np.full(image_array.shape, 255)
    image_array = H - image_array
    self.previewImage = pygame.surfarray.make_surface(image_array)

```

- Dual Channel

Diimplementasikan menggunakan tanpa library selain PyGame. Channel selain channel yang dipilih nilainya diubah menjadi 0.

```

def dual(self, layer):
    image_array = pygame.surfarray.array3d(self.previewImage)
    if layer == 1: # Red Channel
        image_array[:, :, 1] = 0
        image_array[:, :, 2] = 0
    elif layer == 2: # Green Channel
        image_array[:, :, 0] = 0
        image_array[:, :, 2] = 0
    elif layer == 3: # Blue Channel
        image_array[:, :, 0] = 0
        image_array[:, :, 1] = 0
    self.previewImage = pygame.surfarray.make_surface(image_array)

```

- Edge Enhance

Diimplementasikan menggunakan bantuan library Pillow. Citra difilter menggunakan fungsi filter dari module Image dengan parameter `ImageFilter.EDGE_ENHANCE`.

```
def edge(self):
    image_array = pygame.surfarray.array3d(self.previewImage)
    image = Img.fromarray(image_array)
    image = image.filter(ImageFilter.EDGE_ENHANCE)
    self.previewImage = pygame.surfarray.make_surface(np.array(image))
```

- Emboss

Diimplementasikan menggunakan bantuan library Pillow. Citra difilter menggunakan fungsi filter dari module Image dengan parameter `ImageFilter.EMBOSS`.

```
def emboss(self):
    image_array = pygame.surfarray.array3d(self.previewImage)
    image = Img.fromarray(image_array)
    image = image.filter(ImageFilter.EMBOSS)
    self.previewImage = pygame.surfarray.make_surface(np.array(image))
```

- Contour

Diimplementasikan menggunakan bantuan library Pillow. Citra difilter menggunakan fungsi filter dari module Image dengan parameter `ImageFilter.CONTOUR`.

```
def contour(self):
    image_array = pygame.surfarray.array3d(self.previewImage)
    image = Img.fromarray(image_array)
    image = image.filter(ImageFilter.CONTOUR)
    self.previewImage = pygame.surfarray.make_surface(np.array(image))
```

c. Advanced

- Face Detection

Merupakan satu-satunya fitur advanced yang dikembangkan. Dikembangkan menggunakan bantuan library tensorflow. Model face detection di latih menggunakan dataset Wider Face. Terdapat error dan akurasi rendah dalam pengembangan. Selain itu terdapat masalah perbedaan resolusi antara

gambar input dan shape yang diterima model dan masalah lain yang belum selesai hingga tenggat waktu yang diberikan. Oleh sebab itu, fitur face detection tidak dapat dilanjutkan ke tahap implementasi pada aplikasi.

3. Evaluasi Hasil

Pada tahap ujicoba aplikasi, seluruh fitur dites dan di evaluasi kekurangannya. Seluruh filter yang diimplementasikan dapat berfungsi dengan baik, begitu juga dengan open file dan save file. Akan tetapi terdapat perbedaan waktu komputasi pada filter blur. Pada filter ini, diperlukan waktu yang lebih lama untuk komputasi karena filter ini menggunakan pendekatan frekuensi domain. Secara menyeluruh, aplikasi berjalan dengan baik meskipun hanya memiliki 9 dari 15 filter yang telah dipilih.