BAB 2
LANDASAN TEORI

2.1 Penelitian Terdahulu

Beberapa penelitian terdahulu telah membahas klasifikasi gambar anime maupun deteksi gambar buatan AI menggunakan pendekatan *deep learning*. [14] menggunakan arsitektur MobileNetV2 dan MobileNetV3 pada dataset 1.000 gambar (750 latih, 250 uji) yang terdiri dari ilustrasi anime hasil NovelAI dan buatan manusia dari Danbooru2021, memperoleh akurasi 96,8–97,2% dengan presisi sempurna (100%) namun recall 94,3–95,0%. Penulis tersebut secara eksplisit menyatakan bahwa pendekatan tersebut memerlukan pengembangan lebih lanjut dengan dataset berukuran lebih besar untuk dapat diterapkan pada pasar seni daring (*online art marketplaces*). Penelitian ini merespons batasan tersebut dengan memperluas skala dataset menjadi 5.329 gambar dari lima generator AI berbeda dan menggunakan EfficientNetV2-S dengan modul CBAM untuk mengevaluasi apakah kombinasi arsitektur yang lebih dalam dan dataset yang lebih besar dapat meningkatkan konsistensi presisi-recall pada tugas yang serupa.. [23] menerapkan InceptionV3, InceptionResNetV2, MobileNetV2, dan EfficientNet untuk klasifikasi karakter anime, dengan EfficientNet-B7 mencapai akurasi top-1 tertinggi sebesar 85,08%; namun penelitian tersebut berfokus pada identifikasi karakter (klasifikasi multi-kelas), bukan pembedaan asal-usul gambar (AI vs manusia), sehingga hasilnya tidak dapat dibandingkan secara langsung dengan tugas klasifikasi biner pada penelitian ini. Penelitian lainnya [24] mengkombinasikan EfficientNetB0 dengan CBAM untuk mengestimasi tingkat keparahan (severity) penyakit karat pada tanaman gandum, menggantikan modul SE bawaan EfficientNet dengan CBAM untuk menangkap informasi kanal dan spasial secara bersamaan, dan mencapai akurasi pengujian 96,68% pada dataset WheatSev. Hasil ini menunjukkan bahwa CBAM efektif meningkatkan kemampuan model dalam melokalisasi fitur relevan pada citra biologis, namun penelitian tersebut berada pada domain pertanian dan tugas estimasi keparahan multi-tingkat, bukan pada domain ilustrasi anime maupun tugas klasifikasi biner asal-usul gambar (AI vs manusia).

Berdasarkan penelitian-penelitian tersebut, terlihat adanya celah penelitian: (1) pendekatan tanpa mekanisme atensi [14] masih memiliki batas recall akibat kesulitan membedakan fitur halus antar kelas, dan (2) efektivitas kombinasi

EfficientNet dengan CBAM [24] terbukti kuat pada citra biologis namun belum divalidasi pada domain ilustrasi digital yang memiliki karakteristik tekstur dan gaya visual sangat berbeda.

Tabel 2.1. Perbandingan Penelitian Terdahulu

Penelitian	**Metodologi & Hasil**	**Celah yang Diisi Penelitian Ini**
[14]	MobileNetV2/V3 pada dataset 1.000 gambar (NovelAI vs Danbooru); akurasi 96,8–97,2%, presisi 100%, recall 94,3–95,0%.	Memperluas skala dataset menjadi 5.329 gambar dari lima generator berbeda dan menggunakan EfficientNetV2-S dengan CBAM untuk mengevaluasi generalisasi pada dataset besar.
[23]	InceptionV3, InceptionResNetV2, MobileNetV2, EfficientNet-B7 untuk klasifikasi karakter multi-kelas; akurasi top-1 terbaik 85,08%.	Menguji EfficientNet pada tugas biner AI vs manusia, bukan identifikasi karakter.
[24]	EfficientNetB0 + CBAM untuk estimasi keparahan karat gandum; akurasi 96,68% pada dataset WheatSev.	Memvalidasi kombinasi EfficientNetV2 + CBAM pada domain visual berbeda (ilustrasi anime) dan tugas klasifikasi biner, serta menggunakan dataset dari lima generator AI berbeda dengan strategi *source-aware splitting* untuk menguji generalisasi terhadap variasi gaya visual.

2.2 Klasifikasi Gambar

Klasifikasi gambar adalah proses pengelompokan gambar ke dalam kategori atau kelas tertentu berdasarkan karakteristik visual yang ada [25]. Proses ini biasanya dilakukan dengan pendekatan *machine learning* atau lebih spesifiknya, di bidang *computer vision*. Proses ini umumnya dilakukan dengan memanfaatkan model *deep learning* yang dapat mempelajari fitur-fitur secara otomatis dari data gambar [26].

Secara umum, model klasifikasi gambar bekerja dengan mengekstraksi fitur dari gambar *input* melalui rangkaian proses komputasi, lalu memetakan fitur yang dilihat atau diproses oleh model ke dalam kategori yang ada [27]. Proses ini menghasilkan nilai probabilitas dari setiap kategori yang ada, di mana kategori dengan probabilitas tertinggi dipilih sebagai hasil prediksi. Pendekatan ini memungkinkan model untuk mengenali dan menangkap pola dari gambar tanpa memerlukan ekstraksi fitur secara manual.

Dalam implementasi, kinerja dari model klasifikasi gambar sangat dipengaruhi oleh kemampuan model dalam mengekstraksi dan memproses fitur yang relevan dari gambar. Oleh karena itu, pemilihan arsitektur yang tepat dalam mempermudah penangkapan pola visual, seperti bentuk dan tekstur secara efektif

sangat diperlukan. Salah satu arsitektur yang sering digunakan dan memenuhi kebutuhan ini adalah *Convolutional Neural Network* (CNN) [28].

2.2.1 Gambar Anime

Anime merupakan gaya seni yang berasal dari Jepang dan memiliki karakteristik visual yang khas dan berbeda secara signifikan dibandingkan dengan gambar natural atau fotografi. Secara umum, gambar anime ditandai dengan penggunaan *line art* yang tegas dan jelas, pewarnaan *flat shading* dengan saturasi warna yang tinggi, serta proporsi karakter yang khas seperti mata yang besar dan ekspresi yang dilebih-lebihkan [29], [30].

Dalam *computer vision*, gambar anime menghadirkan tantangan tersendiri dalam proses klasifikasi. Beberapa tantangan utama yang dihadapi antara lain:

- **Variasi gaya antar ilustrator**: Setiap ilustrator atau studio animasi memiliki gaya visual yang berbeda-beda, sehingga gambar dari kelas yang sama dapat memiliki penampilan visual yang sangat bervariasi.
- **Minimnya tekstur natural**: Tidak seperti gambar fotografi yang kaya akan tekstur alami, gambar anime cenderung memiliki area warna yang seragam sehingga fitur berbasis tekstur kurang efektif.
- **Dominasi fitur struktural**:Gambar anime lebih mudah dikenali melalui ketegasan *line art*, pilihan warna, dan bentuk karakter secara keseluruhan, dibandingkan melalui detail tekstur seperti yang ditemukan pada gambar natural.

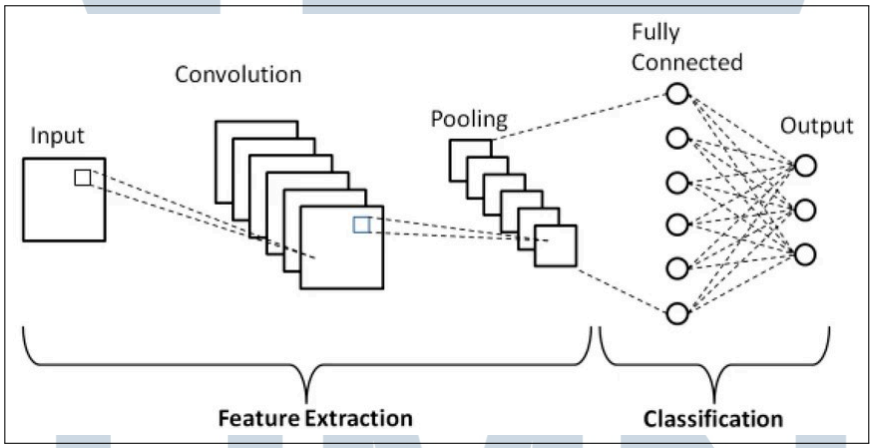
Karakteristik-karakteristik tersebut mempengaruhi cara model *deep learning* mengekstraksi dan memproses fitur dari gambar anime. Oleh karena itu, diperlukan arsitektur model yang mampu menangkap fitur struktural dan spasial secara efektif, sekaligus memiliki kemampuan untuk memfokuskan perhatian pada bagian gambar yang paling relevan untuk klasifikasi.

2.3 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan arsitektur model *deep learning* yang populer dalam bidang *computer vision*, dimana model ini dimanfaatkan dalam tugas seperti deteksi wajah, pengolahan data visual, dan

klasifikasi gambar [31]. Arsitektur CNN sering digunakan karena kemampuan uniknya dalam memproses dan mengolah data visual seperti gambar secara otomatis, sehingga tidak memerlukan *feature engineering* secara manual. Hal ini menjadikannya sangat efektif dan umum digunakan dalam tugas klasifikasi gambar.

Sebelum pendekatan berbasis CNN populer, metode klasifikasi gambar umumnya mengandalkan ekstraksi fitur secara manual (*handcrafted features*) seperti *Histogram of Oriented Gradients* (HOG) dan *Scale-Invariant Feature Transform* (SIFT). Pendekatan tersebut memerlukan keahlian domain yang mendalam dan seringkali tidak mampu menggeneralisasi dengan baik pada variasi data yang tinggi. CNN mengatasi keterbatasan ini dengan mempelajari representasi fitur secara otomatis langsung dari data, sehingga lebih adaptif terhadap variasi visual yang kompleks [32].



Gambar 2.1. Arsitektur CNN konvensional
Sumber: [33]

Secara umum, suatu CNN terkomposisi oleh 3 komponen utama yaitu *convolutional layer*, *pooling layer*, dan *fully connected layer*. *Convolutional layer* adalah lapisan pertama setelah adanya input gambar. Lapisan ini berfungsi untuk mengekstraksi fitur dari gambar dengan menggunakan *filter* atau *kernel* yang digeser ke seluruh bagian dari suatu gambar. Hasil dari lapisan ini kemudian diproses fungsi aktivasi seperti ReLU *(Rectified Linear Unit)* yang befungsi untuk memperkuat hasil dari *convolutional layer* dengan menambahkan non-linearitas ke model sehingga model dapat mempelajari pola yang kompleks. Selanjutnya, *pooling layer* digunakan untuk mengurangi dimensi fitur yang dihasilkan agar dapat menurunkan kompleksitas komputasi serta membantu mengurangi resiko *overfitting*. Setelah melalui beberapa tahap ekstraksi fitur, hasilnya akan diteruskan

ke *fully connected layer* yang berfungsi untuk melakukan klasifikasi berdasarkan fitur yang telah diperoleh. Pada tahap ini, fitur yang telah diekstraksi akan dipetakan ke dalam ruang kategori untuk menghasilkan prediksi akhir.

Keunggulan utama CNN terletak pada kemampuannya dalam menangkap informasi spasial dan pola hierarkis pada gambar. Lapisan awal CNN cenderung mengekstraksi fitur sederhana seperti tepi dan garis, sedangkan lapisan yang lebih dalam mampu mengenali fitur yang lebih kompleks. Hal ini menjadikan CNN pilihan yang populer untuk menangani gambar dengan karakteristik visual yang beragam, termasuk gambar bergaya anime.

2.4 EfficientNet

EfficientNet merupakan arsitektur CNN yang dikembangkan untuk meningkatkan performa model klasifikasi gambar dengan melakukan penskalaan model secara efisien. Pada arsitektur CNN konvensional, peningkatan performa umumnya dilakukan dengan memperbesar dimensi model seperti kedalaman (*depth*), lebar (*width*), atau resolusi *input* (*resolution*) secara terpisah. Pendekatan tersebut seringkali kurang optimal karena dapat menyebabkan peningkatan kompleksitas komputasi yang signifikan, tanpa memberikan peningkatan performa yang cukup.

EfficientNet mengusulkan pendekatan yang lebih sistematis melalui metode yang disebut *compound scaling*, yaitu penskalaan ketiga dimensi tersebut secara bersamaan dengan proporsi tertentu. Dengan pendekatan ini, model dapat mencapai keseimbangan antara akurasi dan efisiensi komputasi.

2.4.1 Compound Scaling

Metode *compound scaling* pada EfficientNet mengatur penskalaan kedalaman, lebar, dan resolusi citra menggunakan suatu konstanta skala ϕ. Hubungan tersebut dapat dinyatakan sebagai berikut:

$$depth = \alpha^{\phi}, \quad width = \beta^{\phi}, \quad resolution = \gamma^{\phi} \tag{2.1}$$

dengan batasan:

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \tag{2.2}$$

dimana α, β, dan γ merupakan konstanta yang ditentukan melalui proses optimasi. Pendekatan ini memungkinkan model untuk meningkatkan kapasitasnya secara seimbang tanpa meningkatkan kompleksitas secara berlebihan.

2.4.2 EfficientNetV2

EfficientNetV2 merupakan pengembangan dari EfficientNet yang dirancang untuk meningkatkan efisiensi pelatihan serta performa model. EfficientNet standar menggunakan blok *Mobile Inverted Bottleneck Convolution* (MBConv) yang memisahkan operasi ekspansi kanal dan konvolusi spasial secara berurutan. EfficientNetV2 mengintroduksi penggunaan *Fused-MBConv*, yaitu kombinasi antara operasi konvolusi dan ekspansi fitur menjadi satu operasi konvolusi yang mempercepat proses pelatihan, terutama pada tahap awal jaringan.

Selain itu, EfficientNetV2 juga mengadopsi strategi penskalaan yang lebih adaptif serta teknik pelatihan yang lebih efisien, termasuk penggunaan ukuran gambar yang lebih kecil pada tahap awal pelatihan dan ditingkatkan secara bertahap (*progressive learning*), sehingga mampu mencapai konvergensi yang lebih cepat dibandingkan EfficientNet [15]. EfficientNetV2 tersedia dalam tiga varian utama, yaitu S (*Small*), M (*Medium*), dan L (*Large*), yang masing-masing menawarkan keseimbangan berbeda antara kapasitas model dan efisiensi komputasi. Pada penelitian ini, digunakan varian EfficientNetV2-S sebagai arsitektur dasar.

2.5 *Attention Mechanism* dan CBAM

2.5.1 Attention Mechanism

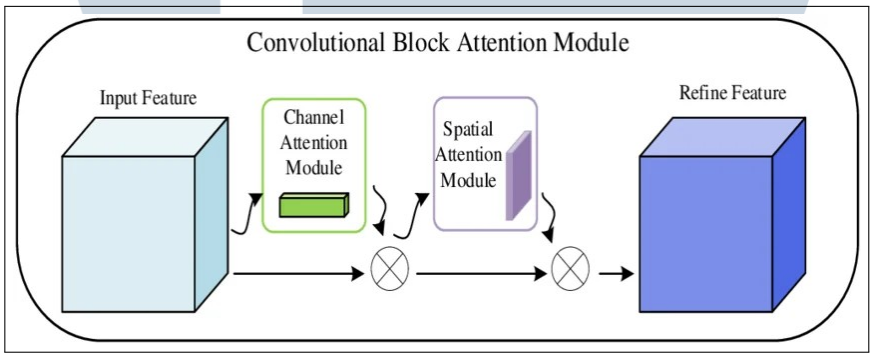
Attention mechanism merupakan konsep dalam *deep learning* yang memungkinkan model untuk memberi fokus spesifik seperti layaknya perhatian pada bagian fitur yang paling relevan dalam suatu data. Dalam proses pengolahan gambar, tidak semua informasi visual dari data memiliki tingkat kepentingan yang sama, sehingga diperlukan mekanisme yang dapat menyesuaikan bobot fitur berdasarkan tingkat relevansinya. Dengan demikian, model dapat mengekstraksi dan fokus terhadap informasi yang lebih penting dan mengabaikan informasi yang kurang signifikan.

Berbeda dengan lapisan konvolusi konvensional yang menerapkan kernel tetap secara seragam ke seluruh feature map tanpa memandang konten input,

attention mechanism menghasilkan bobot yang bersifat *input-dependent*: bobot ini dihitung dari feature map itu sendiri, kemudian digunakan untuk menskalakan (bukan mentransformasi) feature map yang sudah ada.

2.5.2 Convolutional Block Attention Module (CBAM)

Convolutional Block Attention Module (CBAM) merupakan modul perhatian yang dirancang untuk meningkatkan kualitas representasi fitur pada arsitektur CNN. CBAM bekerja dengan menerapkan dua mekanisme perhatian secara berurutan, yaitu *channel attention* dan *spatial attention*. Modul ini dapat diintegrasikan ke dalam berbagai arsitektur CNN sebagai komponen tambahan tanpa meningkatkan kompleksitas komputasi secara signifikan.



Gambar 2.2. Diagram Arsitektur modul CBAM

Sumber: [34]

Secara umum, CBAM menerima masukan berupa *feature map* hasil ekstraksi CNN, kemudian memprosesnya melalui dua tahap perhatian untuk menghasilkan *feature map* yang telah diperkuat. Proses ini bertujuan untuk meningkatkan representasi fitur yang relevan serta menekan fitur yang kurang penting.

2.5.3 Channel Attention

Channel attention berfokus pada penentuan fitur apa yang penting dengan memberikan bobot pada setiap kanal fitur. Proses ini dilakukan dengan menerapkan operasi *global average pooling* dan *global max pooling* pada *feature map*, yang kemudian diteruskan ke jaringan *fully connected* untuk menghasilkan vektor perhatian kanal. Vektor ini digunakan untuk mengalikan *feature map* awal sehingga

kanal yang lebih relevan akan diperkuat. Operasi *pooling* global ini mereduksi dimensi spasial *feature map* berukuran $C \times H \times W$ menjadi vektor perhatian berukuran $C \times 1 \times 1$, sehingga setiap kanal direpresentasikan oleh satu nilai skalar yang mencerminkan tingkat kepentingannya.

2.5.4 Spatial Attention

Spatial attention berfokus pada lokasi atau area mana yang penting dalam suatu gambar. Berbeda dengan *channel attention*, mekanisme ini mengolah informasi spasial dengan menggabungkan fitur antar kanal menggunakan operasi *pooling*. Hasilnya kemudian diproses menggunakan operasi konvolusi untuk menghasilkan peta perhatian spasial (*spatial attention map*) berukuran $1 \times H \times W$, yaitu satu bobot untuk setiap lokasi spasial yang berlaku merata ke seluruh kanal pada lokasi tersebut, yang digunakan untuk memberi fokus ke area penting pada gambar.

2.5.5 Penerapan CBAM pada Penelitian

Pada penelitian ini, CBAM digunakan untuk meningkatkan kemampuan model dalam mengekstraksi fitur visual yang relevan dari gambar anime. Dengan adanya variasi gaya visual dan kompleksitas pola pada gambar anime, penggunaan mekanisme perhatian diharapkan dapat membantu model dalam memfokuskan perhatian pada fitur-fitur penting. Dengan demikian, integrasi CBAM pada arsitektur EfficientNetV2 diharapkan dapat meningkatkan performa klasifikasi gambar.

2.6 Metrik Evaluasi

Metrik evaluasi digunakan untuk mengukur kinerja model dalam melakukan klasifikasi citra. Evaluasi dilakukan dengan membandingkan hasil prediksi model terhadap label sebenarnya pada data uji. Beberapa metrik yang umum digunakan dalam tugas klasifikasi citra antara lain *accuracy*, *precision*, *recall*, dan *F1-score*.

2.6.1 Accuracy

Accuracy merupakan metrik yang mengukur proporsi jumlah prediksi yang benar terhadap seluruh data. Nilai *accuracy* dapat dihitung menggunakan

persamaan berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.3}$$

di mana TP (*True Positive*) adalah jumlah prediksi positif yang benar, TN (*True Negative*) adalah jumlah prediksi negatif yang benar, FP (*False Positive*) adalah jumlah prediksi positif yang salah, dan FN (*False Negative*) adalah jumlah prediksi negatif yang salah.

2.6.2 Precision

Precision mengukur tingkat ketepatan model dalam memprediksi kelas positif. Metrik ini menunjukkan seberapa banyak prediksi positif yang benar dibandingkan dengan seluruh prediksi positif yang dihasilkan oleh model.

$$Precision = \frac{TP}{TP + FP} \tag{2.4}$$

2.6.3 Recall

Recall mengukur kemampuan model dalam menemukan seluruh data yang termasuk ke dalam kelas positif. Metrik ini menunjukkan seberapa banyak data positif yang berhasil dikenali oleh model.

$$Recall = \frac{TP}{TP + FN} \tag{2.5}$$

2.6.4 F1-Score

F1-score merupakan rata-rata harmonis dari *precision* dan *recall*, yang digunakan untuk memberikan keseimbangan antara kedua metrik tersebut.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{2.6}$$

2.6.5 Confusion Matrix

Confusion matrix merupakan representasi tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi dengan menampilkan jumlah prediksi benar dan salah untuk setiap kelas. Matriks ini terdiri dari empat komponen utama, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Dengan menggunakan *confusion matrix*, dapat dilakukan analisis lebih lanjut terhadap kesalahan prediksi yang dilakukan oleh model.

2.6.6 *Receiver Operating Characteristic* (ROC) dan *Area Under Curve* (AUC)

Kurva *Receiver Operating Characteristic* (ROC) adalah alat evaluasi yang menggambarkan performa model klasifikasi pada berbagai nilai batas klasifikasi [35]. Kurva ini dibentuk dengan menempatkan dua metrik dalam suatu *plot* secara bersamaan. Kedua metrik ini adalah *True positive rate* (TPR) yang mengukur seberapa banyak gambar yang benar-benar positif berhasil diprediksi dengan benar dan *False positive rate* (FPR) yang mengukur seberapa sering gambar negatif salah diprediksi sebagai positif. Kedua metrik tersebut dapat didefinisikan sebagai:

$$TPR = \frac{TP}{TP+FN}, \quad FPR = \frac{FP}{FP+TN} \tag{2.7}$$

Area Under Curve (AUC) mengukur luas area di bawah kurva ROC dengan rentang nilai 0 hingga 1. Nilai AUC sebesar 1 menunjukkan pemisahan sempurna antar kelas, sedangkan nilai 0,5 setara dengan klasifikasi acak [35]. Dalam konteks penelitian ini, AUC-ROC digunakan untuk mengevaluasi kemampuan model dalam membedakan gambar anime asli dan buatan AI secara keseluruhan, terlepas dari ambang batas klasifikasi yang dipilih.

2.6.7 *Brier Score*

Brier score adalah metrik pengukuran kualitas prediksi probabilistik model dengan menghitung rata-rate kuadrat selisih antara probabilitas prediksi dan label sebenarnya dengan rumus:

$$BS = \frac{1}{N}\sum_{i=1}^{N}(p_i - y_i)^2 \tag{2.8}$$

di mana N adalah jumlah sampel, p_i adalah probabilitas prediksi model untuk sampel ke-i, dan y_i adalah label sebenarnya (0 atau 1). Nilai *Brier Score* berkisar antara 0 hingga 1, di mana nilai yang lebih rendah menunjukkan kualitas prediksi probabilistik yang lebih baik dengan nilai sempurna adalah 0 [36].

2.6.8 *Expected Calibration Error* (ECE)

Expected Calibration Error (ECE) adalah metrik yang mengukur seberapa jauh tingkat keyakinan (*confidence*) prediksi model mencerminkan akurasi aktualnya [37]. Model yang terkalibrasi dengan baik (*well-calibrated*) idealnya menghasilkan prediksi di mana ketika model menyatakan keyakinan sebesar $x\%$, maka sekitar $x\%$ dari prediksi tersebut memang benar secara aktual.

ECE dihitung dengan membagi seluruh prediksi ke dalam M kelompok (*bin*) berdasarkan tingkat keyakinannya, kemudian menghitung rata-rata tertimbang dari selisih absolut antara akurasi dan keyakinan rata-rata di setiap *bin*:

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{N} |\text{acc}(B_m) - \text{conf}(B_m)| \tag{2.9}$$

di mana $|B_m|$ adalah jumlah sampel dalam *bin* ke-m, N adalah jumlah total sampel, $\text{acc}(B_m)$ adalah akurasi aktual dalam *bin* tersebut, dan $\text{conf}(B_m)$ adalah rata-rata keyakinan prediksi dalam *bin* tersebut. Nilai ECE yang lebih rendah menunjukkan kalibrasi yang lebih baik. Kalibrasi model dapat divisualisasikan menggunakan *reliability diagram*, yaitu grafik yang menggambarkan keyakinan prediksi terhadap akurasi aktual, di mana model yang sempurna terkalibrasi akan menghasilkan garis diagonal sempurna.

2.7 Interpretabilitas Model

Interpretabilitas model merupakan kemampuan untuk memahami dan menjelaskan dasar pengambilan keputusan suatu model *deep learning*, yang menjadi penting terutama pada arsitektur CNN yang bersifat *black-box*. Salah satu pendekatan yang umum digunakan untuk tujuan ini adalah *Class Activation Mapping* (CAM), yaitu teknik visualisasi yang menunjukkan wilayah gambar yang paling berkontribusi terhadap suatu prediksi.

2.7.1 Gradient-weighted Class Activation Mapping (Grad-CAM)

Gradient-weighted Class Activation Mapping (Grad-CAM) merupakan pengembangan dari CAM yang tidak memerlukan modifikasi arsitektur model, sehingga dapat diterapkan pada berbagai arsitektur CNN tanpa pelatihan ulang [38]. Grad-CAM bekerja dengan menghitung gradien dari skor kelas target terhadap peta fitur pada lapisan konvolusi tertentu, umumnya lapisan konvolusi terakhir sebelum *pooling*. Gradien tersebut kemudian dirata-ratakan secara global untuk menghasilkan bobot kepentingan tiap kanal fitur, yang selanjutnya dikombinasikan dengan peta fitur melalui penjumlahan berbobot dan fungsi aktivasi ReLU untuk menghasilkan peta aktivasi (*activation map*). Peta aktivasi ini menunjukkan wilayah gambar yang paling berpengaruh terhadap prediksi model, dengan nilai yang lebih tinggi mengindikasikan kontribusi yang lebih besar.

Dalam penelitian ini, Grad-CAM digunakan untuk membandingkan wilayah gambar yang menjadi fokus perhatian model *baseline* dan model CBAM, guna memahami apakah kedua arsitektur mempelajari representasi fitur yang serupa atau berbeda dalam melakukan klasifikasi gambar anime buatan AI dan manusia.