

BAB 2

LANDASAN TEORI

2.1 Image Processing

Image processing merupakan proses awal yang digunakan untuk citra dengan mengubah citra tersebut menjadi bentuk yang lebih tepat. Contoh operasi yang dapat dilakukan yaitu *color balancing*, *noise reduction*, dan menambah ketajaman gambar (Szeliski, 2010). *Image processing* dapat dibagi dalam tiga kategori, yaitu (Putra, 2010):

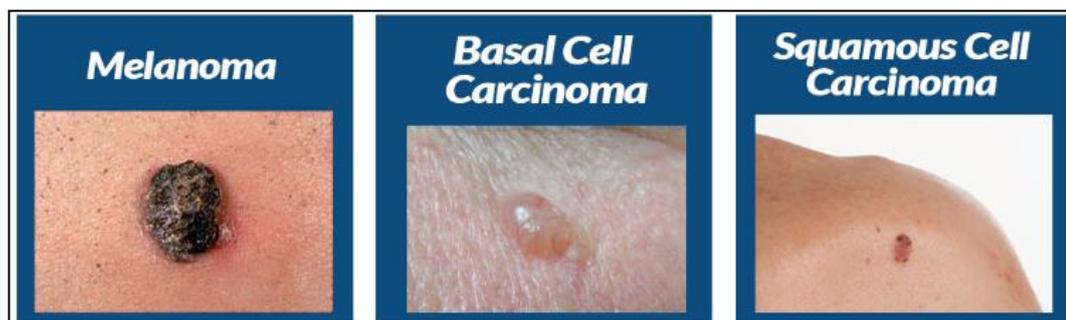
1. Kategori rendah melibatkan operasi-operasi sederhana seperti pra-pengolahan citra untuk mengurangi derau, pengaturan kontras, dan pengaturan ketajaman citra. Pengolahan kategori rendah ini memiliki *input* dan *output* berupa citra.
2. Pengolahan kategori menengah melibatkan operasi-operasi seperti segmentasi dan klasifikasi citra. Proses pengolahan citra menengah ini melibatkan *input* berupa citra dan *output* berupa atribut (fitur) citra yang dipisahkan dari citra *input*. Pengolahan citra kategori melibatkan proses pengenalan dan deskripsi citra.
3. Pengolahan kategori tinggi melibatkan objek-objek yang sudah dikenali menjadi lebih berguna, berkaitan dengan aplikasi, serta melakukan fungsi-fungsi kognitif yang diasosiasikan dengan *vision*.

2.2 Computer Vision

Computer Vision adalah ekstraksi informasi secara otomatis dari gambar. Informasi dapat berarti apa saja dari model 3D, posisi kamera, deteksi dan pengenalan objek untuk mengelompokkan dan mencari konten gambar. Tujuan dari *computer vision* adalah untuk mengerti konten dari gambar. Secara umum, *computer vision* dikembangkan untuk memberikan komputer kemampuan melihat seperti kemampuan penglihatan manusia (Solem, 2012).

2.3 Kanker Kulit

Kanker kulit adalah jenis kanker yang tumbuh di jaringan kulit. Kondisi ini ditandai dengan perubahan pada kulit, seperti munculnya benjolan, bercak, atau tahi lalat dengan bentuk dan ukuran yang tidak normal. Penyebab terjadinya kanker kulit adalah seringnya terkena sinar UV yang berasal dari paparan sinar matahari dan warna kulit. Warna kulit yang cerah cenderung sensitif terhadap sinar matahari dari pada warna kulit yang lebih gelap (Armstrong dan Kricger, 2001).



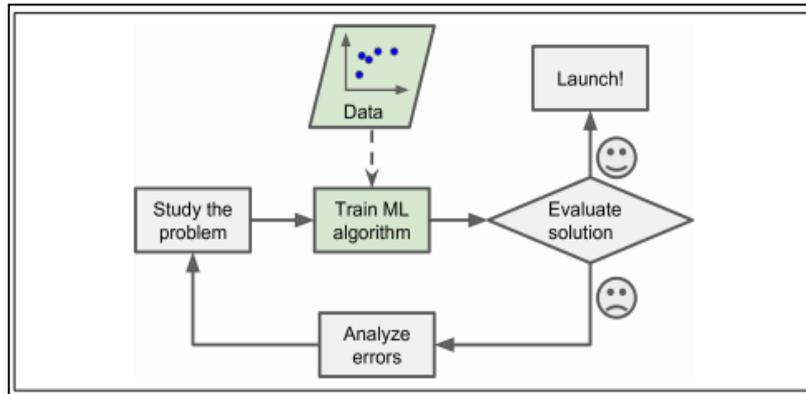
Gambar 2.1 Jenis kanker kulit (Dermatology, 2019)

Terdapat tiga jenis kanker kulit yaitu, *basal cell carcinoma*, *squamosa cell basal*, *melamona malignant* seperti yang ditunjukkan pada Gambar 1 (Hendaria dkk., 2013).

1. *Basal cell carcinoma*, merupakan tipe kanker kulit terbanyak, bersifat lokal invasif, dan lambat untuk menyebar ke bagian kulit lainnya. *Basal cell carcinoma* memiliki bentuk yang menyerupai kutil, tidak berambut, berwarna coklat atau hitam, dan tidak mengkilat (keruh).
2. *Squamosa cell carcinoma*, merupakan tipe kanker kulit yang berasal dari sel skuamosa pada lapisan epidermis kulit. *Squamosa cell carcinoma* memiliki bentuk yang seperti benjolan kecil yang keras.
3. *Melamona malignant*, merupakan tipe kanker kulit yang berasal dari tumor melanosit. *Melamona* memiliki bentuk seperti bercak dengan warna kecoklatan dan terdapat sedikit benjolan di permukaan kulit.

2.4 Pembelajaran Mesin

Pembelajaran mesin adalah bidang studi yang memberikan komputer kemampuan untuk belajar tanpa di program secara eksplisit (Géron, 2019). Untuk menyelesaikan sebuah masalah *machine learning* membutuhkan data sebagai bahan belajar (*training*) dan melakukan evaluasi terhadap model sebelum dapat menghasilkan *output* seperti yang ditunjukkan pada Gambar 2.2.



Gambar 2.2 Proses umum pada *machine learning* (Géron, 2019)

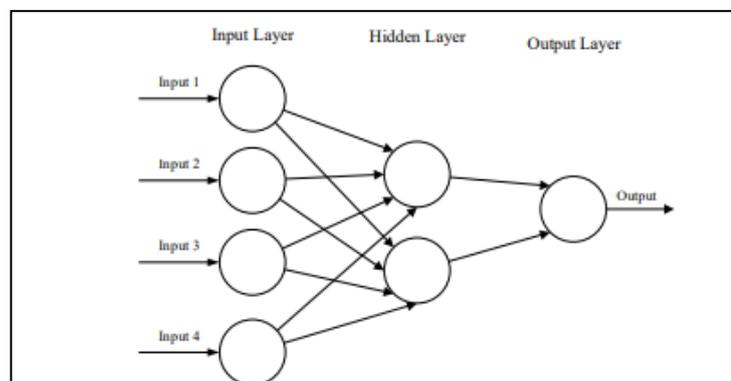
Terdapat tiga jenis *machine learning* berdasarkan teknik pembelajaran yaitu, *supervised learning*, *unsupervised learning*, dan *reinforcement learning* (Géron, 2019).

1. *Supervised learning* adalah pembelajaran mesin dengan menggunakan *training data* yang di beri solusi/label. Contoh tugas yang dapat dilakukan oleh *supervised learning* adalah klasifikasi *spam email*, sistem akan dilatih dengan data yang sudah diberi label (*spam* atau *ham*), dan melakukan klasifikasi terhadap *email* baru.
2. *Unsupervised learning* adalah pembelajaran mesin dengan menggunakan *training data* yang tidak di beri label. Sistem berusaha untuk belajar tanpa ada guru. Contoh tugas yang dapat dilakukan oleh *unsupervised learning* adalah deteksi group dari pengunjung blog, sistem akan membuat beberapa group sesuai dengan kebiasaan dari pengunjung blog.
3. *Reinforcement learning* adalah pembelajaran mesin dengan menggunakan *agent* yang dapat melakukan observasi terhadap lingkungan, pemilihan dan melakukan aksi, dan mendapat penghargaan untuk setiap aksi yang

dilakukan. *Reinforcement learning* banyak digunakan dalam implementasi cara berjalan robot. Setiap kesalahan aksi yang dilakukan oleh robot, sistem akan secara otomatis memberi hukuman sehingga robot tersebut tidak mengulangi aksi tersebut.

2.5 Artificial Neural Network

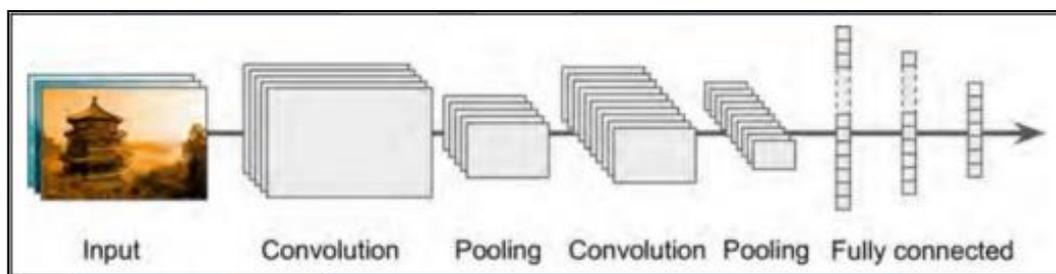
Artificial Neural Network (ANN) merupakan model pembelajaran mesin yang terinspirasi oleh cara sistem saraf manusia (Géron, 2019). Menurut O'shea (2015), *Neural network* terdiri dari sejumlah besar *node* komputasi yang saling berhubungan (*neuron*), yang saling bekerja terkait dalam mode terdistribusi untuk secara kolektif belajar dari *input* untuk mengoptimalkan hasil akhir. Struktur dasar *neural network* dapat dimodelkan seperti Gambar 2.3. Pada struktur tersebut, sistem akan menerima *input* berupa *multidimensional vector* kepada *input layer* yang akan mendistribusikan hasil *input* tersebut ke *hidden layer*. Selanjutnya, *hidden layer* akan membuat keputusan dari *layer* sebelumnya dan membandingkan perubahan stokastik dalam dirinya sendiri merugikan atau meningkatkan hasil akhir (O'Shea dan Nash, 2015).



Gambar 2.3 Struktur dasar *neural network* (O'Shea dan Nash, 2015)

2.6 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan sebuah algoritma *deep learning* yang dapat menerima sebuah *input* berupa citra, dan dapat menetapkan perbedaan setiap gambar secara handal (Saha, 2018). Seperti yang ditunjukkan pada Gambar 2.4, *convolutional neural network* memiliki 4 jenis *layer*.



Gambar 2.4 Arsitektur *convolutional neural network* (Géron, 2019)

1. *Input Layer*

Input Layer merupakan *layer* yang akan menerima *input* berupa citra.

2. *Convolutional Layer*

Convolutional layer merupakan *layer* yang memiliki fokus terhadap *kernel*.

Kernel ini biasanya berukuran kecil dalam dimensi spasial, tetapi menyebar di sepanjang keseluruhan pada kedalaman *input*. Ketika data masuk ke *layer* ini, operasi konvolusi akan dilakukan pada setiap *filter* sepanjang ukuran dimensi spasial dari *input*. Pada proses konvolusi, perkalian skalar akan dilakukan antara bagian kecil dari *input* dengan parameter dari kernel (O'Shea dan Nash, 2015). Ukuran dimensi dari bagian kecil *input* ini sering disebut sebagai *receptive field size*. *Convolutional layer* memiliki rumus sebagai berikut.

$$(f * h)[m, n] = \sum_j \sum_k h[j, k] f[m - j, n - k] \quad (2.1)$$

Pada rumus 1, variabel f merupakan *input image*, variabel h merupakan kernel, variabel m dan n merupakan baris dan kolom (Skalski, 2019). *Output* yang dihasilkan akan dipengaruhi oleh tiga hyperparameter berikut (O'Shea dan Nash, 2015).

- a. *Depth*, merupakan ukuran kedalaman *output*. Mengurangi *depth* dapat secara signifikan mengurangi jumlah *neurons*.
- b. *Stride*, merupakan jumlah pergeseran *filter* terhadap pixel pada *input*.
- c. *Zero-Padding*, merupakan operasi sederhana dengan menambahkan nilai 0 pada pinggiran *input*.

Ukuran pada *output* dapat dirumuskan sebagai berikut.

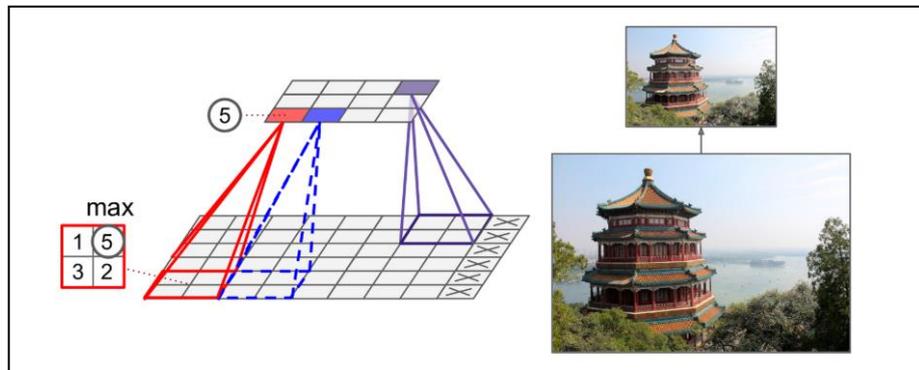
$$n_{out} = \frac{(I-R+2Z)}{S} + 1 \quad (2.2)$$

Pada Rumus 2, variabel n_{out} merupakan hasil ukuran *output*, variabel I merupakan ukuran dari citra, variabel R merupakan bagian citra yang sedang di proses, variabel Z merupakan jumlah *zero-padding*, dan variabel S merupakan *stride* (Skalski, 2019).

3. *Pooling layer*

Pooling layer merupakan *layer* yang berfungsi untuk mengurangi dimensi, dan akan secara otomatis mengurangi jumlah parameter dan kompleksitas model komputasi. Pada *layer* ini, *layer* akan melakukan operasi aktivasi pada setiap bagian pada *input*. Terdapat dua jenis *pooling*, yaitu *max pooling* dan *average pooling*. *Max pooling* mengambil nilai tertinggi dari bagian

input yang sedang di proses. *Average pooling* mengambil nilai rata-rata dari bagian *input* yang sedang di proses (O'Shea dan Nash, 2015).



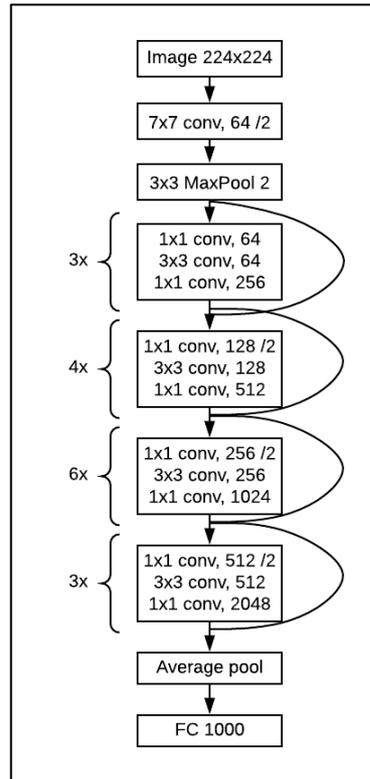
Gambar 2.5 *Pooling layer* (Géron, 2019)

4. *Fully-connected layer*

Fully-connected layer merupakan *layer* terakhir dari CNN. Pada *layer* ini setiap *neuron* terkoneksi terhadap seluruh aktivasi pada *layer* sebelumnya dan *neuron* pada *layer* berikutnya (O'Shea dan Nash, 2015).

2.7 Pretrained Model ResNet50

Pada penelitian ini terdapat arsitektur yang digunakan yaitu arsitektur ResNet50 yang dirancang oleh Microsoft Research. ResNet50 memiliki jumlah *layer* sebanyak 52 *layer*, yang terdiri dari 48 *convolution layer*, 1 *pooling layer*, 1 *average pooling layer*, dan satu *dense layer* (He dkk., 2016). Arsitektur ResNet50 ditunjukkan pada Gambar 2.7. *Pretrained model* yang digunakan sudah dilatih dengan *dataset* ImageNet.



Gambar 2.6 Arsitektur ResNet50

2.8 Confusion Matrix

Confusion matrix adalah matriks yang digunakan untuk melakukan evaluasi performa terhadap klasifikasi sebuah model yang dibuat (Narkhede, 2018). Secara keseluruhan *confusion matrix* dapat dilihat pada Gambar 2.6.

| | | Actual Values | |
|------------------|--------------|---------------|--------------|
| | | Positive (1) | Negative (0) |
| Predicted Values | Positive (1) | TP | FP |
| | Negative (0) | FN | TN |

Gambar 2.7 *Confusion matrix* (Narkhede, 2018)

Menurut Narkhede (2018), pengertian label data pada *confusion matrix* adalah sebagai berikut.

1. TP (*True Positive*), merupakan data yang diprediksi kelas positif dan hasil prediksi benar.
2. TN (*True Negative*), merupakan data yang diprediksi kelas negatif dan hasil prediksi benar.
3. FP (*False Positive*), merupakan data yang diprediksi kelas positif dan hasil prediksi salah.
4. FN (*False Negative*), merupakan data yang diprediksi kelas negatif dan hasil prediksi salah.

Data *confusion matrix* dapat digunakan untuk melakukan perhitungan terhadap *precision*, *recall*, *f1-score* dan akurasi dari sebuah model. Berikut merupakan persamaan yang dapat digunakan untuk mencari *precision*, *recall*, *f1-score* dan akurasi (Géron, 2019).

$$precision = \frac{TP}{TP+FP} \quad (2.3)$$

$$recall = \frac{TP}{TP+FN} \quad (2.4)$$

$$f1 - score = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} \quad (2.5)$$

$$akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.6)$$