Journal of Information System, Graphics, Hospitality and Technology
Volume 03, Nomor 02, Oktober 2021
P-ISSN : 2622-1594
E-ISSN : 2685-449X

Identifikasi Jenis Daging Menggunakan Algoritma Convolution Neural Network

Peter Winardi[a], Endang Setyati[b]
[a,b] *Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya*
E-mail: peter.winardi@gmail.com, endang@stts.edu

***Abstrak*— Kebutuhan protein tubuh manusia salah satunya didapatkan dari daging. Pada kondisi daging mentah, tidak semua orang memahami karakteristik / identitas jenis daging karena ada beberapa jenis daging mentah yang hampir sama dari tampilan visual. Untuk menghindari kesalahan saat pemilihan jenis daging yang diinginkan perlu dilakukan identifikasi jenis daging. Pengenalan jenis daging dapat dilakukan dengan pengambilan gambar / citra secara digital yang didapatkan dapat dilakukan identifikasi dengan *ConvolutionNeural Network*. Pada penelitian ini identifikasi jenis daging yang digunakan berupa adalah daging mentah tanpa lemak, kulit dan tulang. Jenis daging mentah yang digunakan sebanyak 5 buah berupa ayam, babi, bebek, kambing dan sapi. Melalui ekstraksi warna dan deteksi tepi dengan CNN didapatkan identitas jenis daging tersebut berupa tulisan / text sesuai jenis daging input citra. Dataset yang digunakan sebanyak 2,250 citra pada tiap jenis daging sehingga total 11,250 dataset citra. Penelitian dilakukan dalam 2 bagian sistem arsitektur yaitu *Training* dan *Validation* beserta *testing*. Pada bagian training dan validation dilakukan *preprocessing* . citra *resize* dari ukuran 300 × 300 piksel menjadi 50 × 50 piksel. Output training dan validasi berupa penyimpanan konfigurasi CNN yang dihasilkan untuk pemodelan saat testing beserta grafik cross entropy. Pembagian dataset citra model training dan validasi sebesar 70% training dan 30% validasi. Sistem testing digunakan uji coba menentukan jenis daging untuk mendapatkan output tulisan / text dari nama daging yang sesuai. Untuk training dan validasi dilakukan uji coba pertama pada dataset dengan *resize* citra pada ukuran 50 X 50 pixel didapatkan hasil : training loss= 43.89%;training accuracy= 82.82% ; validation loss= 87.44% ; validation juga dilakukan pada ukuran accuracy: 72.27%. Uji coba training dan validasi ke dua dilakukan *resize* citra pada ukuran 100 X 100 pixel dengan hasil : training loss= 35.74% ;training accuracy= 85.75% ; validation loss: 81.08% ; validation accuracy: 71.65%. Uji coba testing didapatkan nilai tertinggi dari angka array hasil pembandaingan dengan penyimpanan konfigurasi training dan validasi. Penelitian identifikasi jenis daging bisa ditingkatkan lebih baik bila dilengkapi dengan dataset citra yang lebih memadai.**

Kata Kunci -Convolution Neural Network, Daging, Identifikasi, Keras, Python, Tensorflow

Naskah Masuk : 23 November 2021
Naskah Direvisi : 01 Desember 2021
Naskah Diterima : 03 Desember 2021

I. PENDAHULUAN

Pemenuhan kebutuhan protein tubuh manusia salah satunya dengan mengkonsumsi daging, diantaranya sapi, kambing, babi, ayam dan bebek. Selain daging sebagai salah satu sumber protein, daging juga mengandung banyak kandungan protein, daging juga memiliki unsur zat besi yang dapat mencegah terjadinya anemia beserta unsur lainnya diantaranya Selenium, Zinc, Vitamin B Komplek dan lainnya[1].Saat ini identifikasi daging dibutuhkan guna membedakan jenis daging saat kondisi mentah. Hal ini dilakukan mengingat ada kasus penipuan dengan mencampurkan daging sapi dengan jenis daging yang lainnya[2]. Tujuan penelitian ini diharapkan dapat membedakan jenis daging dengan menggunakan komputer melalui *computer vision*.

Di dalam penelitian yang sebelumnya, banyak ditemukan identifikasi dan klasifikasi daging maupun klasifikasi kualitas daging yang menggunakan teknologi pengenalan citra pada Computer Vision, diantaranya : Simulasi dan Analisis Pengenalan Citra Daging Sapi dan Daging Babi dengan Metode GLCM[3], *Mobile Application to Differentiate Flesh Meat Between Beef and Pork*[4] dan masih banyak lainnya. Bahkan dalam perkembangan Computer Vision beberapa penelitian sudah banyak menggunakan teknologi Convolution Neural Network untuk klasifikasi, diantaranya *Convolutional Neural Networks for Image Classification*[5], *Computer Vision Detection of Salmon Muscle Gaping Using Convolutional Neural Network Features*[6], *Multi-Task Learning for Food Identification and Analysis with Deep Convolutional Neural Networks*[7], *Fish Detection Using Deep Learning*[8]. Pada Pada penelitian Salmon[6], computer vision digunakan untuk mengidentifikasi seberapa besar luka sayatan hasil fillet yang dilakukan, sedangkan untuk penelitian analisis dan identifikasi jenis makanan[7] berfungsi untuk mengetahui bahan baku yang digunakan melalui gambar / citra dari makanan yang siap saji. Melalui *Computer Vision* diharapkan pula dapat membedakan jenis daging mentah, diantaranya ayam, babi, bebek, kambing dan sapi.

Pada perkembangan penelitian banyak kita temukan pemanfaatan *Machine Learning*. Dalam pemanfaatannya, deep learning merupakan keunggulan yang dimiliki dari *Machine Learning*, khususnya untuk bidang *Computer Vision*. Melalui kemampuannya memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar. Pada Machine Learning terdapat teknik

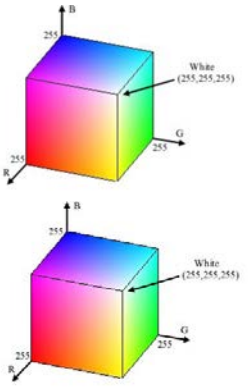
DOI : 10.37823/insight.v3i02.178

untuk menggunakan ekstraksi fitur dari data pelatihan dan algoritma pembelajaran khusus untuk mengklasifikasi citra maupun untuk mengenali suara. Namun, metode ini masih memiliki beberapa kekurangan baik dalam hal kecepatan dan akurasi.

II. TINJAUAN PUSTAKA

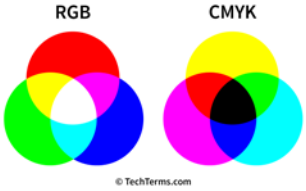
A. *Model Warna (RGB, CMYK, HSV dan Greyscale)*

Salah satu model warna dikenal dengan istilah RGB, dari kepanjangan *Red*, *Green* dan *Blue* (Merah, Hijau dan Biru). Biasanya jenis warna ini dipresentasikan dalam penggunaan monitor komputer dan televisi[9]. Untuk warna yang banyak digunakan di komputer grafik selain RGB [10], ada juga CMYK[11] dan HSI (*Hue, Saturation and Intensity*) atau juga sering disebut HSV (*Hue, Saturation and Value*)[12]. Pemodelan dari warna RGB digambarkan dengan RGB cube dengan sifat warna additive. Pemodelan warna RGB seperti Gambar 1. Di dalam warna RGB terdapat nilai masing-masing komponen warna dengan tingkat warna dari masing-masing *channel red*, *green*, dan *blue* dari 0-255.



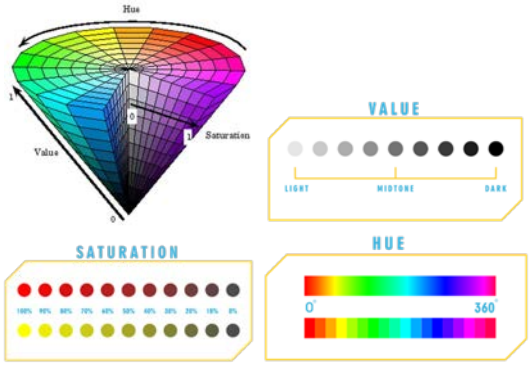
Gambar 1 RGB Cube

Pemodelan dari warna CMYK dengan sifat warna *subtractive*. Pemodelan warna CMYK hampir sama dengan RGB. Perbedaan antara RGB dan CMYK terletak pada warna utama yang digunakan sebagai model. Warna RGB menggunakan model warna utama merah (*red*), hijau (*green*) dan biru (*blue*). Untuk warna CMYK yang digunakan warna utama biru kehijauan (*cyan*), magenta dan kuning (*yellow*), untuk K kepanjangan dari *key* yang berarti kunci merupakan komponen warna hitam sebagai penentu tingkat *brigthness*. Perbedaan segmen warna RGB dan CMYK ditunjukkan pada Gambar 2.



Gambar 2 Perbedaan warna RGB dan CMYK (www.techterm.com)

Untuk pembagian segmentasi warna dengan tingat intensitas ada pada warna HSV. Warna HSV (atau dikenal juga dengan istilah HSI) terdiri dari 3 elemen yaitu *Hue* mewakili warna, *Saturation* mewakili tingkatan dominasi warna, dan *Value* mewakili tingkat kecerahan/intensitas. Dengan demikian metode ini cenderung mendeteksi warna dan tingkat dominasi serta kecerahan dari masing-masing koordinat warna. Untuk memahami struktur segmen warna HSV seperti pada Gambar 3.



Gambar 3 Warna HSV

Nilai dari model segmentasi warna HSV didapatkan dari hasil koversi nilai RGB. Bila dimisalkan nilai *MAX* adalah nilai maksimum warna RGB dan nilai *MIN* sebagai nilai minimum warna RGB maka ketentuan nilai HSV untuk nilai *Hue* dilambangkan *H*, nilai *Saturation* dilambangkan *S* dan nilai *Value* dilambangkan *V* dengan rumus sebagai berikut.

$$H=\begin{cases} \text{Tidak tentu} & \text{jika } MAX = MIN \\ 60^0 \times \frac{G-B}{MAX-MIN} + 0^0 & \text{jika } MAX = R \,\&\, G \geq B \\ 60^0 \times \frac{G-B}{MAX-MIN} + 360^0 & \text{jika } MAX = R \text{ dan } G < B \\ 60^0 \times \frac{B-R}{MAX-MIN} + 120^0 & \text{jika } MAX = G \\ 60^0 \times \frac{R-B}{MAX-MIN} + 240^0 & \text{jika } MAX = B \end{cases} \quad (1)$$

$$S=\begin{cases} 0, & \text{jika } MAX = 0 \\ 1 - \frac{MIN}{MAX} & \text{jika } MAX \neq 0 \end{cases} \quad (2)$$

$$V = MAX \quad (3)$$

Salah satu jenis Extraksi Fitur untuk memanipulasi gambar lebih disukai menggunakan representasi HSI. HSI kepanjangan dari *Hue*, *Saturation* (Saturasi) dan *Intensity* (Intensitas), yang di definisikan sebagai berikut [9]:

- *Hue* : berhubungan dengan warna murni.
  Berdasarkan CIE (Commission Internationale d'Eclairage), Hue adalah atribut dari sensasi virtual berdasarkan daerah yang munculmirip dengan warna merah, kuning, hijau, dan biru atau dua kombinasinya.
- Saturasi : Proporsi warna murni
  Berdasarkan CIE, saturasi adalah tingkat warna dari suatu daerah berdasarkan proporsibrightness-nya. Saturasi dimulai dari warna abu-abu hingga pastel dari warna yang tersaturasi.
- Intensitas : kecerahan (tingkat cerah / tidaknya warna)
  Intensitas (Intensity) adalah pengukuran terhadap suatu interval dari spektrum elektromagnetik darisuatu aliran energi yang diradiasikan atau dikenakan ke permukaan. Pada layar monitor terdapatsuatu voltase yang digunakan atau bertugas untuk mengendalikan intensitas dari komponenwarna

DOI : 10.37823/insight.v3i02.178

HSI sendiri merupakan salah satu ekstraksi fitur warna. Untuk mendapatkan ketiga nilai tersebut, perlu dilakukan konversi ruang warna citra yang semula RGB menjadi HSI. Mencari nilai H pada Rumus 4, nilai S pada Rumus 5 dan nilai I pada rumus 6 rumus sebagai berikut.

$$H = Cos^{-1}\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(R-B)]^{\frac{1}{2}}}\right\} \quad (4)$$

$$S = 1 - \frac{3}{(R+G+B)}\,[min.(R,G,B)I \quad (5)$$

$$I = \frac{1}{3}(R+G+B) \quad (6)$$

Untuk jenis Grayscale adalah rentang warna hitam sampai putih. Konversi warna RGB menjadi Grayscale digunakan untuk pengolahan citra misalnya pencocokan warna dan pola. Bila warna RGB dikonversi langsung menjadi Grayscale maka nilai yang didapat menggunakan Rumus 7 yang digunakan :

$$G = \frac{R+G+B}{3} \quad (7)$$

Dalam realita penerapan konversi RGB ke Grayscale, sebelum di konversi masing-masing chanel / jenis warna dipisahkan terlebih dahulu antara merah (Red), hijau (Green) dan biru (Blue) sehingga masing-masing warna memiliki nilai Grayscale sendiri yang mewakili tingkat kecerahan (*brightness*) dari masing-masing jenis warna.

B. Biner

Citra biner adalah citra dengan setiap piksel hanya dinyatakan dengan sebuah nilai dari dua buah kemungkinan yaitu nilai 0 dan 1[13]. Nilai 0 menyatakan warna putih dan nilai 1 menyatakan warna hitam. Nilai 0 (warna putih) biasanya dipakai sebagai warna latar, sedang nilai 1 (warna hitam) dipakai sebagai warna objek citra. Citra seperti ini bisa digunakan untuk proses segmentasi, dimana memisahkan objek dari latar belakangnya. Model citra ini dipakai dalam pemrosesan citra, salah satunya yaitu untuk kepentingan memperoleh tepi bentuk suatu objek. Garis tepi objek yang dihasilkan dengan metode ini diharapkan lebih jelas.

C. Sobel

Metode pendeteksian tepi adalah salah satu metode segmentasi citra. Segmentasi Citra adalah proses analisa citra yang akan membagi citra menjadi beberapa daerah terpisah untuk analisis lebih lanjut untuk sebuah proses dengan harapan dapat memisahkan objek-objek yang berbeda pada sebuah gambar. Jenis model pendeteksian tepi diantaranya : Metode Robert, Metode Prewitt dan Metode Sobel dan Metode Canny.

Deteksi tepi metode Sobel diperoleh dengan memanfaatkan representasi warna dalam bentuk geometri vector yang merupakan jarak dari ruang vektor 3 dimensi. Rumus 8 merupakan formula mencari jarak pada metode Sobel [14]

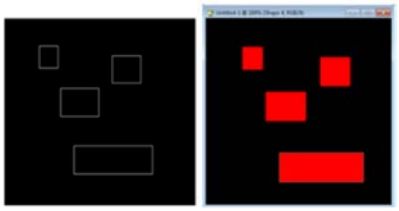
$$D(C1,C2) = \sqrt{(R1-R2)^2 + (G1-G2)^2 + (B1-B2)^2} \quad (8)$$

Dimana D = Jarak warna C1 dan C2
R1, R2= Level warna Merah (Red) dari warna C1 dan C2
G1, G2= Level warna Hijau (Green) dari warna C1 dan C2
B1, B2= Level warna Biru (Blue) dari warna C1 dan C2

Secara umum algoritma pendeteksian tepi adalah sebagai berikut :

a. Ekstrak warna (R,G,B)dari sebuah piksel citra dan simpan hasilnya sebagai C1
b. Ekstrak warna (R,G,B)dari piksel citra di sebelah kanan C1 dan simpan hasilnya sebagai C2
c. Bandingkan nilai warna C1 dan C2
d. Jika perbandingan warna C1 dan C2 melebihi nilai toleransi yang diberikan maka batas antara latar belakang dan objek

Gambar 4 adalah contoh pendeteksian objek pada sebuah citra dengan metode Sobel. Algoritma ini digunakan untuk mengenali posisi dan letak objek di dalam citra agar mudah dideteksi.



(a) (b)

Gambar 4. Pendeteksian Tepi Citra Asli (a) Hasil Deteksi Tepi (b)

Metode Sobel merupakan pengembangan metode Robert dengan menggunakan filter HPF (*High Pass Filter*) yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi *laplacian* dan *gaussian* yang dikenal sebagai fungsi untuk membangkitkan HPF. Kelebihan dari metode Sobel ini adalah kemampuan untuk mengurangi noise sebelum melakukan perhitungan deteksi tepi. Permasalahan pengurangan noise dilakukan dengan tujuan untuk menentukan objek yang diinginkan saja. Untuk perhitungan Sobel menggunakan metode Gradient[13] dengan Arah Gradien digunakan pada rumus berikut ini.

$$\blacktriangledown f = \left|\frac{df}{dx}\right| + \left|\frac{df}{dy}\right| = |Gx| + |Gy| \quad (9)$$

Sedangkan arah gradien :

$$\text{Tan } \alpha\,(x,y) = \frac{Gy}{Gx} \;;\; \alpha\,(x,y) = \tan^{-1}\frac{Gy}{Gx} \quad (10)$$

Untuk filter sobel yang digunakan matrix 3x3 sesuai matrik pada rumus berikut ini.

$$Gx = \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix} \qquad Gy = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \quad (11)$$

Bila diketahui sebuah image Z memiliki nilai sebagai berikut :

$$\begin{matrix} Z1 & Z2 & Z3 \\ Z4 & Z5 & Z6 \\ Z7 & Z8 & Z9 \end{matrix} \quad (12)$$

Maka untuk perhitungan Gx dan Gy dengan filter matrik 3x3 didapatkan perhitungan Rumus sebagai berikut.

$$Gx = -Z1-Z2-Z3+Z7+2Z8+Z9 \quad (13)$$
$$Gy = -Z1+Z3-2Z4+2Z6+-Z7+Z9$$

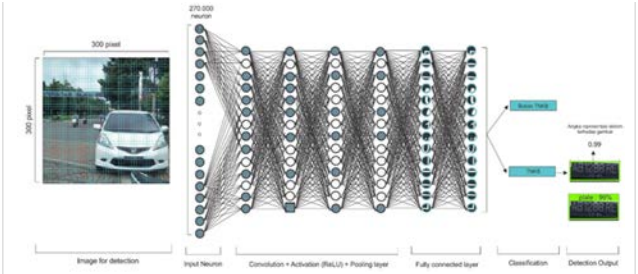
Contoh hasil citra pada deteksi tepi dengan pemakaian matrix 3x3 Sobel pada Gambar 5.

DOI : 10.37823/insight.v3i02.178

Gambar 5. Contoh Citra Deteksi Tepi Dengan Matrix 3x3 Sobel

D. Convolution Neural Network

Untuk memahami arsitektur Convolution Neural Network, lebih dahulu pahami arsitektur neural network yang digunakan untuk pengenalan plat nomor mobil seperti pada Gambar 6.



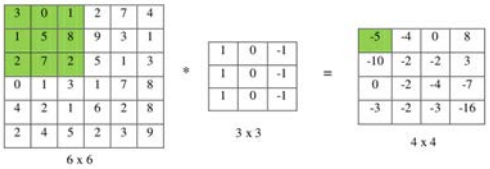
Gambar 6 Arsitektur Neural Network Pada Proses Pengenalan Plat Nomor Mobil

Pada arsitektur di atas terdapat 3 layer secara umum yaitu input layer, hidden layer dan output layer. Berikut ini adalah penjelasan dari masing-masing layer tersebut :

1. **Input layer** merupakan seluruh data yang diperoleh dari gambar pada **image for detection.***Input layer* merupakan seluruh *node* piksel gambar. Pada gambar tersebut terbagi menjadi 300 x 300 piksel, sehingga *node* yang terdapat pada *input layer* sebanyak 90.000 *node*, akan tetapi keterangan pada arsitektur tersebut adalah 270.000.000 yang diperoleh dari 90.000 *node* dari piksel asli yang dikalikan dengan 3 channel warna RGB (*Red, Green, Blue*) sehingga menghasilkan node sebanyak 270.000.000.
2. **Hidden layer** merupakan bagian pada neural network yang tidak ditampilkan, sehingga dinamakan sebagai *hidden layer*. Pada *hidden layer* ini dilakukan proses perhitungan. Dalam hal ini perhitungan dilakukan untuk melakukan deteksi tanda nomor kendaraan. Pada *hidden layer* ini sebenarnya terdapat tiga proses untuk mendapatkan hasil deteksi tanda nomor kendaraan. Tiga proses tersebut adalah proses *convolution*, proses *activation* dan proses *pooling* layer.
3. **Output layer** merupakan hasil dari proses dari *input layer* dan *hidden layer* yang berupa *fully connected layer* atau dengan kata lain adalah meng-gabungkan semua layer yang diperkirakan sebagai tanda nomor kendaraan.

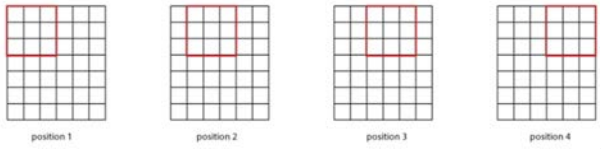
Konvolusi (*convolution layer*) merupakan cara untuk mengkombinasikan dua buah deret angka yang menghasilkan deret angka yang ketiga. Dalam penelitian ini, dua buah deret angka tersebut terdapat dalam *input* dan kernel atau filter sedangkan deret angka yang ketiganya adalah *output*. *Input* dan kernel atau filter keduanya memiliki deret angka berupa matriks. Pada input deret angka diperoleh berdasarkan tingkat warna yang ada pada masing-masing piksel sedangkan pada kernel atau filter deret angka disesuaikan oleh kebutuhan peneliti. Terdapat beberapa jenis kernel atau filter yang biasa digunakan, diantaranya pada operasi *identity*, *edge detection*, *sharpen*, *box blur*, *gaussian blur* dan lain sebagainya.

Lapisan konvolusi dibentuk dengan menjalankan sebuah filter. Filter tersebut merupakan sebuah blok atau kubus dengan tinggi dan lebar yang lebih kecil namun kedalaman yang sama yang tersapu di atas citra / gambar dasar atau gambar asli. Filter digunakan untuk menentukan pola apa yang akan dideteksi yang selanjutnya dikonvolusi atau dikalikan dengan nilai pada matriks input, nilai pada masing-masing kolom dan baris pada matriks sangat bergantung pada jenis pola yang akan dideteksi.



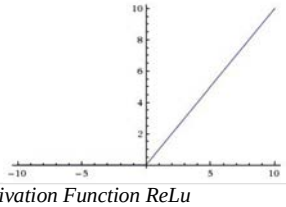
Gambar 7 Sampel Deret Angka Dengan Input Ukuran 6x6 Dan Menggunakan Filter "V*ertical Edge Detection*" Ukuran 3x3

Untuk dapat lebih memahami cara kerja dari proses konvolusi, peneliti menggunakan sampel deret angka pada input dikarenakan ukuran 300x300 maka digunakan sampel deret angka pada input dengan ukuran 6x6 dan menggunakan kernel atau filter untuk operasi *vertical edge detection* dengan ukuran 3x3 sesuai pada Gambar 7



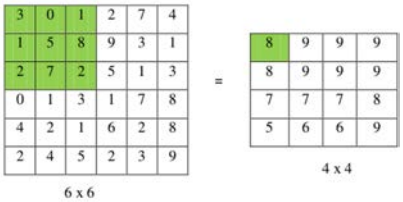
Gambar 8 Proses Konvolusi Pada Filter 3x3

Dengan menggunakan filter 3x3 dengan stride atau langkah yang digunakan dalam perhitungan konvolusi tersebut adalah 1 maka proses perhitungan konvolusi tersebut dapat divisualisasikan seperti Gambar 8. Perhitungan pada proses konvolusi dengan ukuran filter 3x3 ini dimulai dari sudut kiri atas kemudian dilakukan *sliding window* sampai pojok kiri bawah secara merata ke semua bagian. Pada tahap selanjutnya adalah tahap activation layer, dimana dilakukan perhitungan nilai dengan *Activation Function* yang digunakan untuk mencari nilai non-linier pada nilai hasil konvolusi. Secara umum, *Activation function* dapat dibagi menjadi 2 tipe yaitu fungsi aktivasi linier dan fungsi aktivasi non linier. Ada beberapa *activation function* yang akan kita temukan dalam prakteknya, diantaranya Sigmoid, Tanh, ReLU. Untuk penggunaan activation lebih banyak pada ReLU karena sangat mempercepat proses konvergensi yang dilakukan dengan stochastic gradient descent jika dibandingkan dengan sigmoid / tanh.



Gambar 9 *Activation Function ReLu*

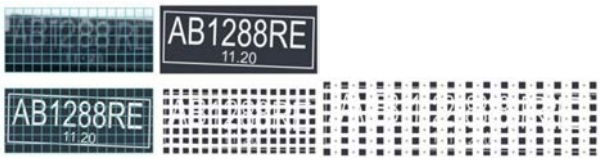
 DOI : 10.37823/insight.v3i02.178



Gambar 10 *Max Pooling Layer*

Pooling layer pada Gambar 10 di atas merupakan pooling layer dengan menggunakan metode *Max Pooling*. *Pooling layer* sendiri berguna untuk mengurangi ukuran gambar. Pada Gambar 10 di atas terdapat lapisan dengan ukuran 6×6, apabila peneliti menggunakan filter 3x3 dengan stride 1 maka diperoleh hasil *Max Pooling* dengan ukuran 4x4.

Fully Connected Layer pada dasarnya menggunakan lapisan yang terhubung sepenuhnya di mana setiap piksel biasa. Lapisan terakhir yang terhubung sepenuhnya akan mengandung banyak neuron sebagai jumlah kelas yang harus diprediksi. Pada proses fully connected layer ini penyatuan setiap piksel yang dianggap sebagai Tanda Nomor Kendaraan yang kemudian akan menjadi sebuah output yang terdiri dari satu label kelas yaitu Tanda Nomor Kendaraan Bermotor sehingga lapisan yang terhubung terakhir hanya memiliki 1 neuron. Untuk hal tersebut bisa dilihat pada Gambar 11.



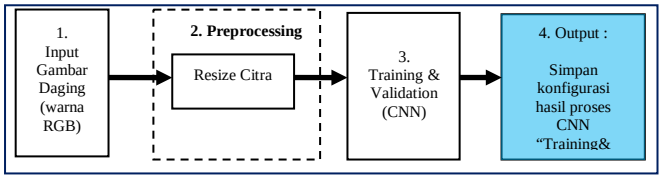
Gambar 11 Contoh Penerapan Fully Conected Layer Pada Plat Nomor Mobil

III. PERANCANGAN SISTEM

Pada penelitian ini menggunakan algoritma CNN. Sistem dibagi dua (2) bagian besar, bagian pertama training dan validasi beserta bagian ke dua testing. Untuk tiap bagian memiliki sistem arsitektur terpisah.

A. *Training dan Validation*

Untuk training dan validasi memiliki sistem arsitektur sebagai berikut :

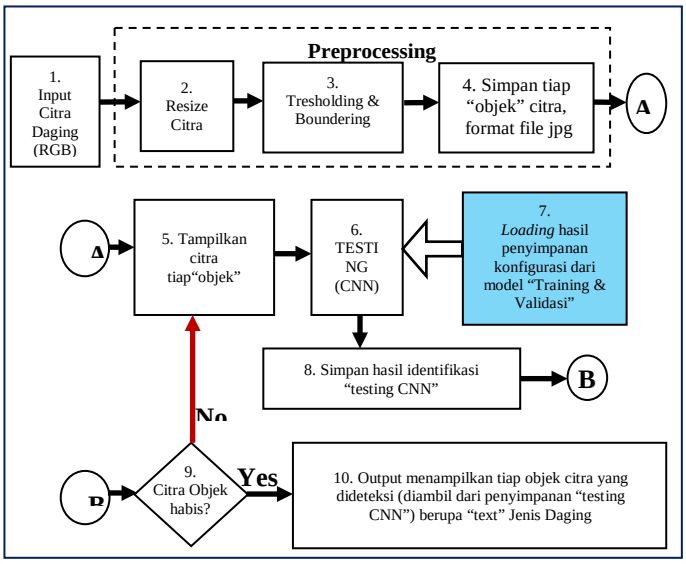


Gambar 12. Diagram sistem arsitektur Training & Validation

Berikut keterangan diagram arsitektur Gambar 12 :

1. Dataset citra RGB dari daging mentah ayam, babi, bebek, kambing dan sapi sebagai input sistem. Semua dataset citra yang akan digunakan dalam format file jpeg dengan ukuran 300 × 300 piksel.
2. Pada saat masuk ke sistem, dataset citra dilakukan resize dengan ukuran 50 x 50 piksel. Pra-proses (preprocessing) citra dilakukan terlebih dahulu untuk penyesuaian ukuran citra. Tujuan resize untuk efisiensi dan efektifitas pada sistem input CNN.
3. Hasil citra *preprocessing* dilakukan proses pada Convolution Neural Network (CNN). Pada proses ini model training dan validasi terbentuk.
4. Output dari proses training dan validasi CNN berupa model konfigurasi hasil training dan validasi CNN. Output ini disimpan untuk model saat Testing.

B. *Testing*



Gambar 13. Diagram sistem arsitektur Testing

Berikut keterangan diagram arsitektur Gambar 13 :

1. Dilakukan proses pengambilan gambar daging mentah untuk Citra Testing. Gambar dengan model warna RGB dalam format jpeg.
2. Identifikasi banyak objek pada citra dilakukan proses *tresholding* dan *boundering*.
3. Citra dilakukan resize dengan ukuran 50 × 50 piksel..
4. Penyimpanan tiap citra objek pada file yang berbeda.
5. Objek Citra yang telah disimpan masing-masing dikeluarkan (*upload*) satu per satu sebagai input CNN.
6. Citra dari proses no. 5 dilakukan testing CNN.
7. Pada saat dilakukan testing CNN, hasil dari training dan validasi CNN di masukkan sebagai model pembanding.
8. Hasil testing CNN disimpan dalam bentuk informasi identitas daging sesuai input citra objek.
9. Apakah semuanya citra objek yang disimpan telah diproses testing CNN? Bila objek citra masih ada dilakukan pengulangan proses pada no. 4 untuk identifikasi objek citra berikutnya. Bila objek citra sudah habis akan masuk proses selanjutnya.
10. Sistem berakhir dengan output / hasil akhir sistem testing akan memberikan tulisan / "text" nama jenis daging sesuai input citra dari masing-masing objek citra yang ada.

IV. HASIL UJI COBA DAN ANALISA PENELITIAN

A. *Dataset*

Gambar 14. Dataset Citra Daging Mentah

Pada penelitian ini digunakan dataset dengan 5 kelas dari jenis daging mentah, yaitu ayam, babi, bebek, kambing dan

DOI : 10.37823/insight.v3i02.178

sapi. Karakteristik gambar daging yang digunakan dalam keadaan mentah serta tidak mengandung lemak, tulang dan kulit. Format gambar dalam bentuk RGB, channel Red, channel Blue, channel Green, channel Magenta (Greyscale) dalam format file JPEG dengan dimensi 300x300 piksel. Jumlah gambar untuk 1 jenis daging sebanyak 2,250 citra, total gambar yang dijadikan objek penelitian sejumlah 11,250 citra. Untuk uji training sebanyak 70 % dari total gambar dan untuk validasi sebanyak 30 % dari total gambar.

B. Pengujian Sistem

Untuk bagian training dan validasi beserta bagian testing digunakan bahasa pemrogaman Python 3.8 dengan aplikasi PyCharm 2020.3.2 (Community Edition).

Uji Training dan Validasi dilakukan dengan membagi dataset citra sebesar 70 % untuk training dan 30 % untuk validasi Untuk uji coba testing digunakan citra training dan validasi beserta citra baru.

Pada bagian training dan validasi dilakukan 2 model uji coba. Uji coba pertama dilakukan preprocessing dengan mengubah (resize) ukuran citra dari 300 X 300 piksel menjadi ukuran 50 X 50 piksel, sedangkan uji coba ke dua dengan mengubah (resize) ukuran citra dari 300 X 300 piksel menjadi 100 X 100 piksel.

Sistem training dan validasi dimulai dari mempersiapkan dataset citra berukuran 300 x 300 pixel sebanyak 2,250 citra tiap jenis daging, total dataset 11,250 citra. Dataset citra daging dipanggil sesuai tempat file dari masing-masing folder gambar. Setelah dipanggil, ukuran citra dilakukan resize pada ukuran 50 X 50 pixel dan 100 X 100 pixel. Pada proses ini dataset citra diubah menjadi data array dengan 2 jenis data. Data X.pickle untuk nilai feature gambar dan data Y.pickle untuk label kelas citra. Data ini disimpan untuk proses CNN. Pada uji coba training dan validasi dilakukan 150 epoch didapatkan hasil :

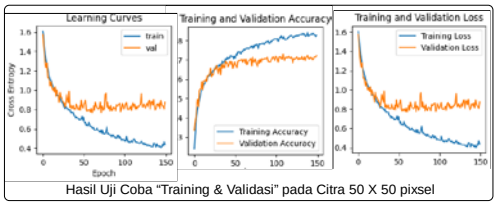
TABEL I
HASIL UJI COBA TRAINING& VALIDASI

Uji Coba	Ukuran Citra (Pixel)	Hasil ke	Training Loss (%)	Training Accuracy (%)	Validasi Loss (%)	Validasi Accuracy (%)	Keterangan
1	50 X 50	150	43.89	**82.82**	87.74	**72.27**	Hasil Akhir
2		109	47.61	**81.04**	79.10	**73.05**	Hasil Max.
3	75 X 75	150	39.86	**83.63**	78.71	**72.07**	Hasil Akhir
4		100	48.26	**80.89**	72.27	**72.93**	Hasil Max.
5	100 X 100	150	35.74	**85.75**	81.08	**71.65**	Hasil Akhir
6		138	35.79	**85.42**	79.08	**72.69**	Hasil Max.

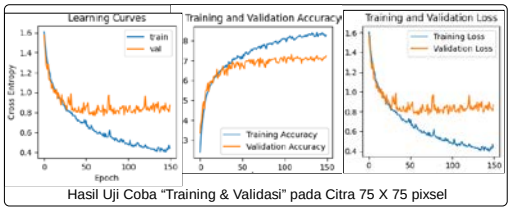
Catatan : Hasil max didapatkan pada nilai akurasi validasi tertinggi

Pada proses CNN, X.pickle dan Y.pickle dipanggil untuk dilakukan normalisasi. Setelah normalisasi dilakukan proses CNN. Algoritma CNN yang digunakan : model sequensial dengan sistem convolusi 2D sebanyak 3 kali masing-masing sebesar 64, 128, 128 beserta kernel size 3 X 3, aktivasi LeakyRelu - alpha = 0.001, Maxpooling ukuran 3 X 3 dan fully connected - dense 2 kali sebesar 64 dan 5.

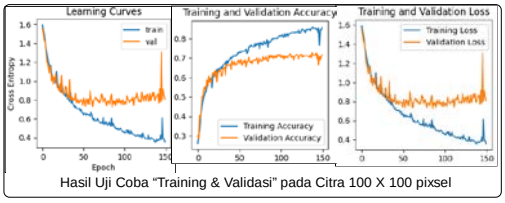
Hasil testing dilakukan pengecekan citra test dengan membandingkan hasil penyimpanan konfigurasi training & validasi yang dibuat sebelumnya. Citra testing sebelumnya juga dilakukan resize sesuai dengan ukuran dari konfigurasi image size yang digunakan (size 50 pixel , 75 pixel atau 100 pixel) dan diubah dalam bentuk data array (untuk pendeteksian fitur). Output testing berupa nilai prediksi (*prediction*) berupa angka array dari tiap kelas dan text sesuai kelas yang cocog dengan melihat angka yang terbesar dari output angka. Berikut hasil output pada uji coba testing dengan menggunakan konfigurasi training & validasi pada ukuran citra 50 pixel, 75 pixel dan 100 pixel berupa grafik akurasi dan error. Untuk hasil testing didapatkan dengan membaca angka yang dihasilkan array pada nilai tertinggi. Semakin besar ukuran citra yang diproses pada sistem CNN didapatkan hasil maksimum lebih tinggi.



Gambar 15. Hasil uji coba pada training dan validasi dengan 150 Epoch dengan ukuran citra 50 X 50 Pixel



Gambar 16. Hasil uji coba pada training dan validasi dengan 150 Epoch dengan ukuran citra 50 X 50 Pixel



Gambar 17. Hasil uji coba pada training dan validasi dengan 150 Epoch dengan ukuran citra 100 X 100 Pixel

Tabel II
Output Testing

Uji Coba	Size (pixel)	Gambar	Output angka	Output Text
1	50	 kambing_08585.jpg	[[-18.365023 -12.986191 -1.2871011 **13.541866** -9.848406]]	Kambing
2	75		[[-18.254765 -4.8603153 4.3431168 **15.066143** -5.1149707]]	Kambing
3	100		[[-16.785105 -4.4801397 4.480502 **17.811594** -9.083682]]	Kambing

V. KESIMPULAN

1. Hasil akhir uji coba "training & validasi" pada ukuran citra 50 X 50 Pixel dengan 150 epoch didapatkan :Training Loss : 43.89 % ; Training Accuracy : 82.82 % ; Validation Loss : 87.74 % ; Validation Accuracy : 72.27 %
2. Hasil maksimum uji coba "training & validasi" pada ukuran Citra 50 X 50 Pixel dengan 150 epoch didapatkan: Training Loss: 47.61 % ; Training Accuracy: 81.04 % ; Validation Loss: 79.10% ; Validation Accuracy: 73.05 % (pada 109 Epoch)

DOI : 10.37823/insight.v3i02.178

3. Hasil akhir uji coba “training & validasi” pada ukuran citra 100 X 100 Pixel dengan 150 Epoch didapatkan :Training Loss : 35.74 % ; Training Accuracy : 85.75 % ; Validation loss : 81.08 % ; Validation Accuracy : 71.65 %
4. Hasil maksimum uji coba “training & validasi” pada ukuran citra 100 X 100 Pixel dengan 150 iterasi didapatkan :Training Loss: 35.79 % ; Training Accuracy: 85.42 %; Validation Loss: 79.08 % ; Validation Accuracy: 72.69 % (pada 138 Epoch)

DAFTAR PUSTAKA

[1] K. Adi, “Detection of the Beef Quality,” *Proc. 2016 3rd Int. Conf. Inf. Tech., Comput. Electr. Eng. (ICITACEE), Oct 19-21st, 2016, Semarang, Indones. Detect.*, vol. 3rd, pp. 253–259, 2016.

[2] I. Wahyudiyanta, “Polisi Bongkar Penjualan Daging Sapi Dicampur Daging Babi di Surabaya,” *May 26, 2016, 14:17*, 2016. http:/news.detik.com/berita-jawa-timur/3218411/polisi-bongkar-penjualan-daging-sapi-dicampur-babi-di-surabaya (accessed Feb. 06, 2019).

[3] S. A. Wibowo, B. Hidayat, and U. Sunarya, “Simulasi dan Analisis Pengenalan Citra Daging Sapi dan Daging Babi dengan Metode GLCM,” *Semin. Nas. Inov. DAN Apl. Teknol. DI Ind. 2016 ISSN*, no. ISSN : 2085-4218, pp. 338–343, 2016.

[4] W. Muhammadiy and F. Fahmi, “Mobile application to differentiate flesh meat between beef and pork,” *Proc. - Cybern. 2016 Int. Conf. Comput. Intell. Cybern.*, vol. 978, pp. 47–50, 2016, doi: 10.1109/CyberneticsCom.2016.7892565.

[5] N. Jmour, S. Zayen, and A. Abdelkrim, “Convolutional Neural Networks for Image Classification,” in *2018 International Conference on Advanced Systems and Electric Technologies, IC_ASET 2018*, Jun. 2018, pp. 397–402, doi: 10.1109/ASET.2018.8379889.

[6] J. L. Xu and D. W. Sun, “Computer Vision Detection of Salmon Muscle Gaping Using Convolutional Neural Network Features,” *Food Anal. Methods*, vol. 11, no. 1, pp. 34–47, 2018, doi: 10.1007/s12161-017-0957-4.

[7] X. J. Zhang, Y. F. Lu, and S. H. Zhang, “Multi-Task Learning for Food Identification and Analysis with Deep Convolutional Neural Networks,” *J. Comput. Sci. Technol.*, vol. 31, no. 3, pp. 489–500, 2016, doi: 10.1007/s11390-016-1642-6.

[8] S. Cui, Y. Zhou, Y. Wang, and L. Zhai, “Fish Detection Using Deep Learning,” *Appl. Comput. Intell. Soft Comput.*, vol. 2020, p. 13 pages, 2020, doi: 10.1155/2020/3738108.

[9] D. Putra, *Pengolahan Citra Digital*, Maret 2014., no. April. Lampung: Perpustakaan Nasional RI, 2010.

[10] T.-T. T. T. C. Dictionary, “RGB,” *TechTerms*. https://techterms.com/definition/rgb.

[11] T.-T. T. T. C. Dictionary, “CMYK,” *TechTerms*. https://techterms.com/definition/cmyk.

[12] M. Galer and L. Horvat, *Digital Image, Essential Skills*, 3rd ed., vol. 53, no. 9. Italy: Elsevier, 2005.

[13] R. E. Gonzalez, Rafael C . ; Woods, *Digital Image Processing (2nd Edition)*, 2nd ed., vol. 14, no. 3. www.prenhall.com/gonzalezwoods, 2014.

[14] R. A. Junior, Nurhasanah, and I. Sanubary, “Perbandingan Penggunaan Beberapa Metode Deteksi Tepi pada Pengolahan Citra Radiologi Fraktur Tulang,” *Prism. Fis.*, vol. 5, no. 3, pp. 117–121, 2017.

 DOI : 10.37823/insight.v3i02.178