

Klasifikasi Buah Zaitun Menggunakan *Convolution Neural Network*

Toton Dwi Antoko^{1*}, Muhammad Azhar Ridani², Agus Eko Minarno³

^{1,2,3}Program Studi Informatika, Fakultas Teknik, Universitas Muhammadiyah Malang
Jl. Raya Tlogomas No. 246 Tlogomas, Malang, Indonesia 65144

*email: totonantoko@gmail.com

(Naskah masuk: 26 Januari 2021; diterima untuk diterbitkan: 10 April 2021)

ABSTRAK – Buah zaitun merupakan tanaman produk hortikultura rumpun oleaceae dengan genus *Olea* yang memiliki berbagai macam jenis dan fitur yang unik. Satu dari sekumpulan species *Olea* yang ditemukan di wilayah tropis dan subtropis yang menjadikan tanaman subur dan melimpah. Hasil panen yang sangat melimpah sebanding dengan kebutuhan pasar, sehingga mengakibatkan pemanenan produk secara acak tanpa mempertimbangkan untuk mengelompokkan jenisnya. Oleh sebab itu, pemilihan produk pasca panen sangat penting dalam mengelompokkan jenis buah zaitun. Sehingga perlu adanya sistem yang dapat mengklasifikasi secara otomatis. Sebelumnya sudah ada penelitian yang diusulkan untuk mengklasifikasi buah zaitun dengan akurasi yang cukup tinggi. Namun kecepatan yang diperlukan butuh waktu yang sangat lama karena menggunakan model pretrained yang begitu kompleks. Oleh karena itu, penelitian ini bertujuan untuk melakukan klasifikasi buah zaitun dengan waktu yang lebih cepat dan akurasi yang tidak kalah dari sebelumnya. Penelitian sebelumnya menggunakan metode dengan model Convolutional Neural Network (CNN) pretrained, sedangkan penelitian ini akan menggunakan metode dengan model Convolutional Neural Network (CNN) dengan rangkaian arsitektur sendiri. Hasil dari penelitian ini mendapatkan akurasi sebesar 92% dengan waktu pelatihan 164 detik dengan 30 epoch.

Kata Kunci – Klasifikasi; Deep Learning; Computer Vision; CNN; Buah Zaitun.

Olive Fruit Classification Using Convolution Neural Network

ABSTRACT – Olive fruit is a horticultural product of the oleaceae family with the genus *Olea* which has various types and unique features. One of a group of *Olea* species found in tropical and subtropical regions which make the plant fertile and abundant. The yields are very abundant in proportion to market needs. The random harvest of produce makes the selection of post-harvest products very important in classifying types of olives. So it is necessary to have a system that can classify automatically. Previous studies have been proposed to classify olives with considerable accuracy. However, the required speed takes a very long time because it uses a complex pretrained model. Therefore, this study aims to classify olives with a faster time and accuracy that is no less than before. The method to be used is Convolutional Neural Network (CNN) with its own architectural circuit. The results of this study get an accuracy of 92% with 30 epochs.

Keywords - Classification; Deep Learning; Computer Vision; CNN; Olive Fruit.

1. PENDAHULUAN

Buah zaitun merupakan tanaman yang termasuk ke dalam keluarga *oleaceae* dengan genus *Olea*. Satu dari sekumpulan species *Olea* yang ditemukan di wilayah tropis dan subtropis, hanya *Olea europaea* L atau buah zaitun yang menghasilkan buah yang dapat dikonsumsi[1]. Pertumbuhan zaitun saat ini merupakan suatu aktivitas yang tumbuh dengan pesat dalam pengembangan budidaya buah. Dengan kecocokan di wilayah Mediterania, penanamannya telah menyebar ke seluruh dunia dalam beberapa

dekade terakhir[2], dan konsumsi buah zaitun yang digunakan menjadi minyak merupakan produk terpenting yang berasal dari tanaman ini. Menurut *International Olive Council (IOC)*, konsumsi buah zaitun dan minyak zaitun telah meningkat lebih dari dua kali lipat dalam 20 tahun terakhir[2]. Oleh karena itu, seperti yang terjadi pada tanaman lainnya, sektor zaitun menghadapi tantangan untuk memenuhi permintaan pasar yang tinggi, di mana popularitas produk turunan zaitun tidak berhenti berkembang. Meningkatkan produksi serta mengurangi beban biaya produksi merupakan masalah dalam sektor

pertanian saat ini[2]. Pengenalan teknologi baru dapat memainkan peran yang mendasar untuk menghadapi situasi ini.

Dengan tidak membedakan varietas pada saat mengumpulkan buah zaitun, biaya panen berpotensi dapat dikurangi. Dalam hal ini, proses pemilihan pascapanen akan menjadi bagian yang paling penting. Ini karena proses pemilihan pascapanen dilakukan secara manual sehingga akan sangat memakan waktu dan cenderung tidak akurat. Oleh karena itu, otomatisasi menjadi masalah utama dalam sebuah industri apa pun. Kasus sektor zaitun tidak terkecuali, karena klasifikasi buah pasca panen tetap menjadi tantangan bagi para petani zaitun.

Beberapa penelitian tentang klasifikasi buah sudah banyak dilakukan. Namun sayangnya untuk klasifikasi buah zaitun masih kurang menarik perhatian bagi para peneliti. Hal itu dapat dilihat pada ketersediaan paper topik ini terlihat masih sedikit. Penelitian yang dilakukan oleh Asia Kausar, dkk tahun 2018 untuk klasifikasi buah menggunakan PureCNN dengan *Global Average Pooling* mendapatkan akurasi sebesar 98% dengan waktu komputasi 20.02 ms. Data yang digunakan pada penelitian tersebut bersumber dari website kaggle.com[3]. Penelitian lain dilakukan oleh Febian Fitra Maulana dan Naim Rochmawati mengusulkan metode CNN arsitektur sendiri dengan memadukan aktivasi ReLu dan Max Pooling Layer. Penelitian ini berhasil mendapatkan akurasi sebesar 97,97% serta data yang digunakan bersumber dari website kaggle.com[4]. Penelitian lain dilakukan oleh Shawon Ashraf, dkk tahun 2019 untuk klasifikasi buah menggunakan *pretrained* model InceptionV3 untuk kategori *multiclass* dan VGG16 untuk kategori *binary class* dengan hasil akurasi masing-masing 87.08% dan 95.08%. Data yang digunakan bersumber dari website Google Search Image[5]. Penelitian oleh M. Ragini, dkk tahun 2020 menggunakan *pretrained* model InceptionV3 untuk klasifikasi buah dan sayuran serta gizinya. Data yang digunakan dikumpulkan sendiri oleh penulis[6]. Penelitian oleh Al-Shawwa Mohammed O tahun 2019 melakukan klasifikasi buah apel menggunakan *deep learning* dengan arsitektur jaringan sendiri mendapatkan hasil akurasi *training* 99.99% dan *testing* 100%. Data yang digunakan bersumber dari website kaggle.com[7]. Penelitian oleh Yudong Zhang, dkk tahun 2019 melakukan klasifikasi buah berdasarkan jenis menggunakan *deep learning* dengan arsitektur sendiri berjumlah 13 layer serta menggunakan data *augmentation* berhasil mendapatkan akurasi sebesar 99.43% dan presisi sebesar 95.34%. Data yang digunakan bersumber dari Google Image Dataset[8]. Penelitian oleh Jose Luis Rojas-Aranda, dkk tahun 2020 melakukan klasifikasi buah untuk toko eceran menggunakan *pretrained* model MobileNetV2

dengan gambar *grayscale* mendapatkan akurasi 93% ketika buah di dalam plastik dan 95% ketika buah di luar plastik. Data yang digunakan bersumber dari website kaggle.com[9]. Penelitian oleh M. Shamim Hossain tahun 2018 melakukan klasifikasi buah secara otomatis menggunakan *deep learning* dengan VGG16 dan arsitektur model sendiri berhasil mendapatkan akurasi masing-masing sebesar 99.75% dan 99.49%. Data yang digunakan bersumber dari internet dan dikumpulkan sendiri oleh peneliti[10].

Penelitian sebelumnya yang dilakukan Juan M. Ponce tahun 2017 melakukan klasifikasi jenis buah zaitun berdasarkan rata-rata menggunakan beberapa *pretrained* model yaitu AlexNet, InceptionResNetV2, InceptionV1, InceptionV3, ResNet50, ResNet101 yang berhasil mendapatkan akurasi masing-masing sebesar 89%, 91%, 94%, 95%, 94, dan 95%. Data yang digunakan merupakan data *private* yang dikumpulkan sendiri. Permasalahan yang didapatkan dari penelitian tersebut terdapat pada waktu komputasi melakukan *training*. Hal itu karena model arsitektur yang digunakan memiliki kompleksitas yang tinggi dan membutuhkan perangkat keras yang bagus[2].

Berdasarkan permasalahan yang telah dijelaskan di atas pada penelitian ini diusulkan klasifikasi buah zaitun menggunakan CNN. Penelitian ini bertujuan untuk mengklasifikasikan buah zaitun dengan waktu komputasi yang lebih cepat. Perbedaan penelitian ini dengan penelitian sebelumnya adalah menggunakan arsitektur model sendiri. Dengan menggunakan arsitektur model sendiri ini diharapkan dapat memperkecil waktu komputasi dan mendapatkan akurasi bagus.

2. METODE DAN BAHAN

Bagian ini mengurai tentang pengumpulan dataset, *pre-processing*, *augmentasi*, *training* dan evaluasi model.

2.1. Pengumpulan Data

Pada tahap ini akan dilakukan pengumpulan data yang dibutuhkan untuk penelitian. Pengumpulan data dilakukan agar memperoleh informasi untuk tercapainya penelitian. Data yang digunakan pada penelitian ini merupakan data yang dikumpulkan sendiri dari *Google Search*. Jumlah data yang didapatkan sebanyak 136 gambar dengan 6 kategori yaitu Arbequina, Arbosana, Changlot, Lechin, Picual dan Verdial. Masing-masing kategori memiliki jumlah gambar yang berbeda-beda.

2.2. Preprocessing Data

Setelah data yang dibutuhkan untuk penelitian sudah terkumpul, maka tahap selanjutnya perlu

dilakukan *preprocessing data*. Dalam setiap gambar dari data yang didapatkan terdapat beberapa gambar buah zaitun dalam satu citra. Oleh karena itu, pada setiap gambar akan dilakukan pemotongan untuk mengambil gambar individu dari buah zaitun. Cara ini akan dilakukan secara manual.

Setiap gambar yang dilakukan pemotongan akan menghasilkan beberapa gambar buah zaitun individu yang berbeda-beda. Hal itu karena di setiap gambar memiliki gambar zaitun individu yang berbeda. Hasil dari tahapan ini mengubah jumlah data yang awalnya berjumlah 136 menjadi 267 citra.

2.3. Augmentasi Data

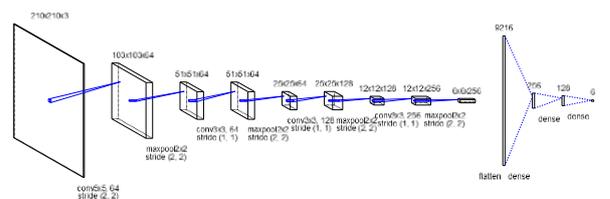
Data yang dikumpulkan dari proses sebelumnya berjumlah sangat sedikit sekali. Meskipun setelah dilakukan proses *preprocessing data* ada sedikit penambahan jumlah citra. Oleh sebab itu, pada tahapan ini akan dilakukan proses augmentasi data untuk memperbanyak data yang digunakan. Augmentasi data ini menggunakan *library tensorflow ImageDataGenerator* dengan paramater *rotation_range=360*. Dengan penggunaan *rotation 360* tersebut akan membuat setiap gambar secara *random* merotasi antara 0 sampai 360 derajat.

2.4. Arsitektur Model

Pada tahapan ini dilakukan penentuan model CNN yang sesuai untuk mengklasifikasi gambar buah zaitun. CNN sendiri merupakan salah satu algoritma dari *deep learning* yang merupakan pengembangan dari *Multilayer Perceptron (MLP)* untuk mengolah data dalam bentuk array matrix berupa dua dimensi atau lebih, seperti data gambar, suara maupun video[11]. CNN digunakan untuk mengklasifikasikan data berlabel menggunakan metode *supervised learning*. CNN sering digunakan untuk mengenali objek atau pemandangan, serta juga untuk melakukan deteksi dan segmentasi objek.

Penentuan arsitektur model pada setiap masalah atau kasus pada project computer vision bisa jadi atau juga bisa berbeda. Selain itu, juga bisa menggunakan arsitektur model yang telah ada dan sudah dilatih sebelumnya. Namun, pada penelitian ini akan dibangun model arsitektur sendiri yang sesuai dengan masalah yang akan diselesaikan.

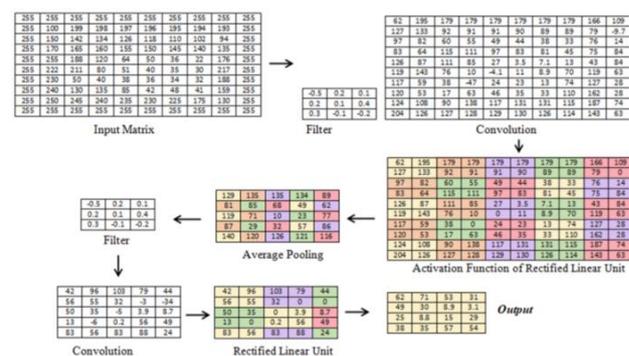
Arsitektur model yang dibangun terdiri dari beberapa *layer* antara lain, *convolution*, *max-pooling*, serta *fully-connected*. Pada Gambar 1 ditunjukkan arsitektur model CNN yang digunakan dan dibangun sendiri.



Gambar 1. Arsitektur Convolution Neural Network (CNN)

2.4.1 Convolution

Convolution merupakan lapisan yang menjadi dasar dari arsitektur CNN untuk melakukan operasi konvolusi pada *output* dari layer sebelumnya[12]. Citra yang dimasukkan ke dalam model dan melewati lapisan ini akan dilakukan proses ekstraksi fitur. Citra tersebut akan di pecah-pecah menjadi beberapa bagian dari setiap *pixel* sesuai dengan parameter yang sudah ditentukan. Dari proses ini, citra bisa menjadi ukuran lebih kecil atau bisa juga dengan ukuran tetap sama, namun kedalaman gambar akan berubah. Cara kerja lapisan *convolution* dapat dilihat pada Gambar 2.



Gambar 2. Proses Konvolusi

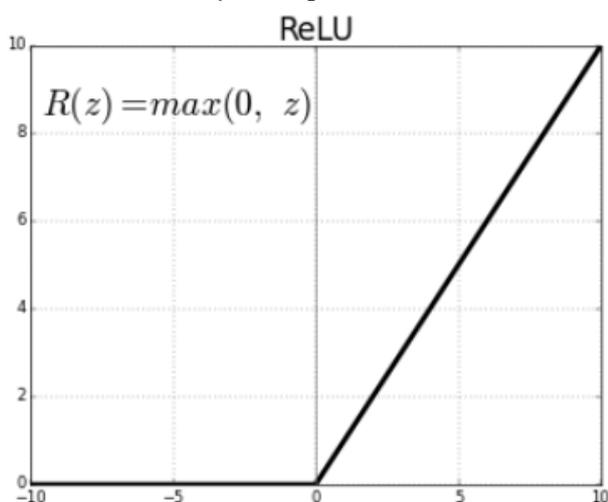
Pada Gambar 2, terdapat sebuah citra berukuran 10x10 piksel digambarkan sebagai matriks. Citra tersebut diproses dengan dua layer konvolusi untuk menghasilkan sebuah feature map. Pada layer konvolusi pertama, filter yang digunakan berukuran 3x3 dengan bobot yang telah ditentukan. Hasil dari konvolusi pertama berupa matriks dengan ukuran 9x9. Setelah melewati konvolusi yang pertama, pada hasil konvolusi feature map akan diberikan fungsi aktivasi. Fungsi aktivasi yang akan digunakan adalah reLu. Hasil dari fungsi aktivasi reLu akan diberikan pada pooling dengan ukuran filter 2x2 serta stride adalah dua. Hasil ini kemudian akan dilakukan konvolusi kedua dengan ukuran filter sama seperti sebelumnya, akan tetapi menggunakan bobot yang berbeda. Dalam hal ini, ukuran tidak harus sama dengan konvolusi tahap pertama dan merupakan parameter yang bisa dioptimalkan. Sementara bobot dari matriks merupakan nilai yang dicari melalui proses pelatihan model.

Hasil dari proses konvolusi kedua akan diberikan dengan fungsi aktivasi yang sama, yaitu reLu.

Pooling yang dikenakan juga filter berukuran 2x2 menggunakan stride satu, sehingga menghasilkan matriks dengan ukuran 4x4. Proses konvolusi bisa dilanjutkan sesuai dengan matriks akhir yang diharapkan. Dalam hal ini, apabila konvolusi tidak dilanjutkan sampai tahap kedua, maka matriks berukuran 4x4 tersebut menjadi input bagi neural network. Apabila filter yang digunakan sejumlah N, maka masukkan bagi jaringan syaraf adalah Nx4x4. Penggunaan fungsi aktivasi dan pooling bisa dirubah urutannya tanpa mempengaruhi hasil dari konvolusi. Perubahan ukuran ini bertujuan untuk mengurangi proses yang digunakan sehingga model menjadi lebih cepat.

2.4.2 Fungsi Aktifasi

Proses penerapan fungsi aktifasi terjadi setelah proses konvolusi dan sebelum terjadi proses pooling. Fungsi aktivasi digunakan untuk membuat model jaringan syaraf tidak menghasilkan nilai non-linear[13]. Pada proses ini, hasil dari konvolusi diberikan fungsi aktivasi. Terdapat beberapa fungsi yang sering digunakan pada layer konvolusi, antara lain *tanh* atau *relu*. Aktivasi relu menjadi pilihan bagi beberapa peneliti karena sifatnya yang lebih berfungsi dengan baik. Fungsi yang digunakan untuk aktivasi pada relu adalah nilai output dari neuron yang bisa dinyatakan sebagai 0 jika inputnya adalah negatif. Jika nilai input dari fungsi aktivasi adalah positif, maka output dari neuron adalah nilai input aktivasi itu sendiri[13]. Grafik fungsi dari aktivasi relu ditunjukkan pada Gambar 3.



Gambar 3. Grafik fungsi aktivasi relu

Selain itu juga terdapat fungsi aktivasi yang cukup penting untuk digunakan di setiap akhir dari model yaitu fungsi sigmoid dan softmax. Fungsi sigmoid akan menerima angka yang berdiri sendiri dan mengubah nilainya menjadi nilai yang memiliki rentang antara 0 dan 1. Fungsi ini, biasanya digunakan pada model untuk klasifikasi yang hanya

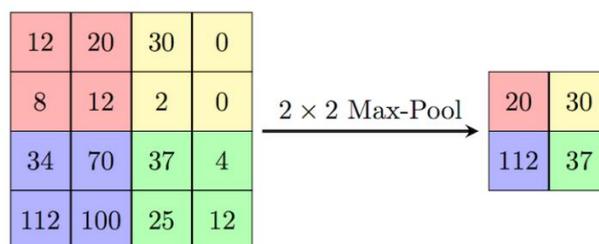
memiliki maksimal dua kelas saja. Sedangkan untuk softmax digunakan pada model untuk melakukan klasifikasi dua kelas atau lebih[13].

2.4.3 Pooling

Pooling adalah pengurangan ukuran matriks dengan menggunakan operasi *pooling*[14]. *Pooling layer* biasanya digunakan setelah *convolution layer*. Proses konvolusi dan *pooling* dilakukan beberapa kali untuk mendapatkan fitur maps dengan ukuran yang dikehendaki. Fitur maps tersebut akan menjadi *inputan* bagi *Fully Connected Layer* atau *dense* yang sudah diubah menjadi array satu dimensi untuk melakukan klasifikasi.

Pada dasarnya pooling layer terdiri dari sebuah filter dengan ukuran dan stride tertentu yang bergeser pada seluruh area pada feature map. Pooling yang biasa digunakan adalah *Max Pooling* dan *Average Pooling*. Tujuan dari penggunaan pooling layer adalah mengurangi dimensi dari feature map atau bisa disebut juga sebagai *downsampling*. Hal tersebut akan mempercepat komputasi karena parameter yang harus di update semakin sedikit dan juga akan mengatasi masalah *overfitting*.

Hal terpenting dalam pembuatan model CNN adalah dengan memilih banyak jenis lapisan pooling. Hal ini akan sangat menguntungkan kinerja model[15]. Lapisan pooling bekerja di setiap kumpulan feature map dan mengurangi ukurannya. Bentuk dan ukuran lapisan pooling yang paling sering digunakan adalah dengan menggunakan filter berukuran 2x2 yang digunakan dengan langkah sebanyak 2 kali dan kemudian beroperasi pada setiap irisan dari input. Bentuk seperti ini akan mengurangi feature map hingga 75% dari ukuran aslinya. Proses *pooling* yang digunakan adalah *MaxPooling*. *MaxPooling* ini akan mengambil nilai maksimum dari setiap filter proses *pooling*. Cara kerja dari lapisan *max pooling* dapat dilihat pada Gambar 4.



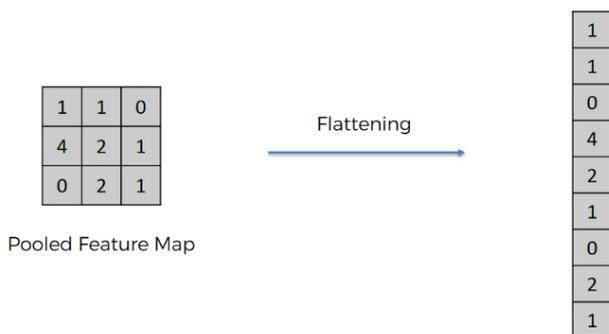
Gambar 4. Proses *Pooling*

Lapisan pooling akan beroperasi pada setiap kedalaman volume *feature map* hasil konvolusi secara bergantian. Pada Gambar 4, lapisan pooling menggunakan salah satu operasi maksimal yang merupakan operasi yang paling umum. Operasi tersebut akan mengambil nilai maksimum dari *feature maps* dengan lebar ukuran sesuai filter yang digunakan. Dari ukuran *feature maps* 4x4, pada

masing-masing empat angka akan diambil nilai maksimalnya dan membuat ukuran output baru menjadi 2x2.

2.4.4 Flatten

Flatten memiliki arti yaitu perataan. Perataan disini ditujukan pada sebuah matrix atau feature maps dari hasil proses konvolusi untuk diubah ukuran menjadi bentuk *array* satu dimensi[16]. Flatten ini menjadi bagian penting bagi model dalam melakukan klasifikasi. Jika tidak dilakukan proses *flatten*, maka hasil dari proses konvolusi tidak bisa diproses oleh *fully-connected-layer*. Cara kerja dari *flatten* seperti yang ditunjukkan pada Gambar 5.

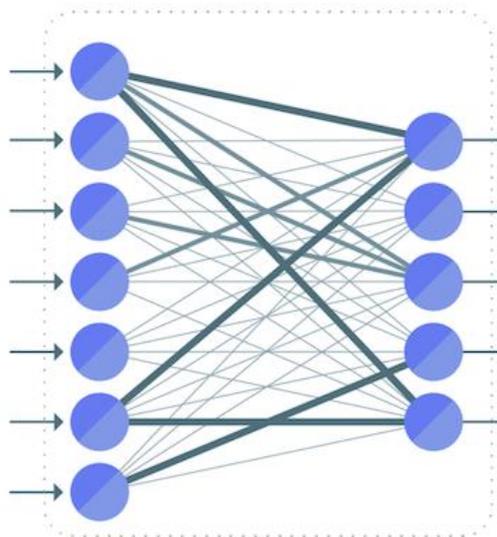


Gambar 5. Proses flatten

2.4.5 Fully Connected Layer

Fully connected layer merupakan lapisan yang terdiri dari beberapa neuron yang terhubung antara satu dengan yang lainnya seperti halnya jaringan syaraf manusia[17]. Neuron-neuron yang membentuk FCN juga sering disebut sebagai dense. Setiap aktivitas yang terjadi pada lapisan sebelumnya harus di jadikan array atau larik data satu dimensi sebelum dapat diproses oleh neuron pada Fully Connected Layer ini.

Lapisan fully connected umumnya digunakan untuk mengolah data supaya model bisa melakukan klasifikasi[17]. Perbedaan antara lapisan fully connected dengan lapisan konvolusi adalah neuron pada konvolusi hanya terhubung pada daerah inputan tertentu. Sedangkan lapisan fully connected layer mempunyai neuron yang saling terhubung secara keseluruhan. Akan tetapi, keduanya masih tetap sama dalam melakukan operasi matrix yaitu dengan menggunakan perkalian dot. Untuk bentuk dari *fully connected layer* seperti yang ditunjukkan pada Gambar 6.



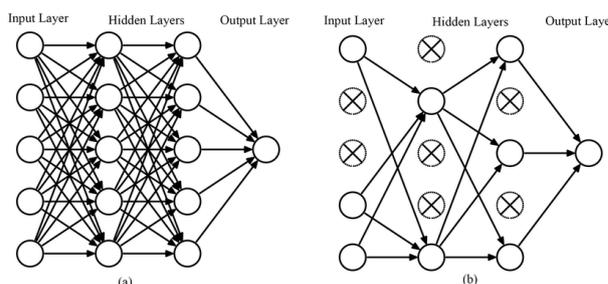
Gambar 6. Bentuk *fully connected layer*

2.4.6 Dropout

Dropout merupakan teknik regularisasi jaringan syaraf dimana beberapa neuron akan dipilih secara acak dan tidak akan digunakan selama proses pelatihan model jaringan. Neuron-neuron tersebut akan diabaikan secara acak. Hal tersebut akan membuat neuron yang diabaikan akan dihentikan sementara oleh model jaringan. Selain itu, hasil bobot yang baru tidak akan diterapkan pada neuron tersebut. *Dropout* akan melakukan proses mencegah terjadinya *overfitting*[16]. Selain itu, *dropout* akan mengurangi kompleksitas atau jumlah parameter yang digunakan dari suatu model sehingga akan mempercepat proses pelatihan model.

Menghilangkan suatu neuron artinya akan menghilangkannya sementara dari jaringan yang ada. Setiap neuron akan diberikan probabilitas yang memiliki nilai antara 0 dan 1. Nilai tersebut bisa ditentukan sebelumnya ketika memberikan dropout pada *hidden layer*.

Pada Gambar 7 dapat dilihat perbedaan antara penggunaan model jaringan syaraf biasa tanpa menggunakan *dropout* dan model jaringan syaraf yang menggunakan *dropout* dimana ada beberapa neuron yang tidak digunakan lagi.



Gambar 7. Cara kerja *dropout*

2.5. Evaluasi Confusion Matrix

Tahapan terakhir setelah model sudah dilatih adalah evaluasi. Evaluasi di sini akan mengukur kinerja model yang melakukan klasifikasi buah zaitun. Salah satu metode yang digunakan untuk mengukur performa dari suatu model klasifikasi adalah dengan menghitung nilai akurasi, precision, serta recall dari suatu model. Beberapa istilah yang sering digunakan dalam melakukan perhitungan kinerja model klasifikasi adalah positive tuple dan negative tuple. Positive tuple adalah jenis tuple yang menjadi titik pembahasan atau nilai yang benar. Sedangkan negative tuple adalah jenis tuple yang tidak menjadi titik fokus pembahasan atau nilai yang salah.

Beberapa istilah lain yang merupakan dasar dalam melakukan perhitungan performa nilai akurasi, precision, dan recall adalah menggunakan parameter true positive (TP), true-negative (TN), false positive (FP), dan false negative (FN)[18]. Istilah-istilah dari parameter tersebut sering disebutkan dalam suatu matriks yang terdapat pada confusion matrix. Setiap nilai dari parameter tersebut diperoleh dari perhitungan confusion matrix seperti pada Tabel 1. Untuk rumus perhitungan dari performa matrix dapat di lihat pada persamaan (1), (2) dan (3).

Tabel 1. Confusion Matrix

		Nilai Sebenarnya	
		Negative	Positive
Nilai Prediksi	Negative	True (TN)	False (FP)
	Positive	False (FN)	True (TP)

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100\% \quad (1)$$

$$recall = \frac{TP}{FP + TP} * 100\% \quad (2)$$

$$precision = \frac{TP}{FN + TP} * 100\% \quad (3)$$

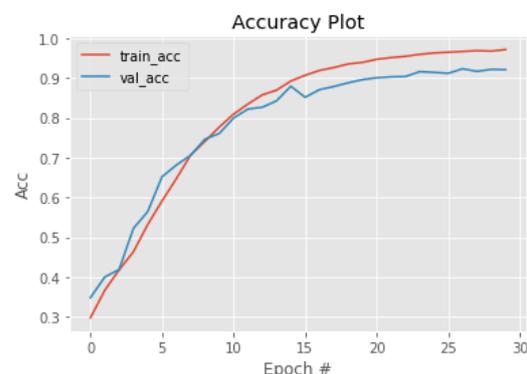
Pada Tabel 1 ditunjukkan, bahwa nilai true positive diartikan sebagai positive tuple yang diklasifikasikan dengan hasil yang benar oleh model. Kedua, true negative adalah negative tuple yang diklasifikasikan dengan nilai yang benar oleh model. Ketiga, untuk false positive adalah negative tuple yang diklasifikasikan sebagai kelas bernilai positif oleh model. Terakhir, untuk false negative adalah positive tuple yang diklasifikasikan sebagai kelas bernilai negatif oleh model klasifikasi.

3. HASIL DAN PEMBAHASAN

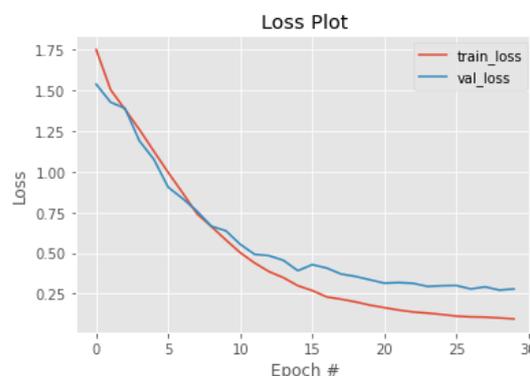
Hasil dari proses training atau pelatihan dalam CNN adalah terbentuknya sebuah model dan record

siap pakai untuk pemeriksaan lebih lanjut atau dengan kata lain disebut sebagai validation ataupun testing. Pada setiap proses training, terdapat dua metrik penilaian yaitu nilai accuracy dan loss. loss merupakan penalti atau hukuman bagi sistem untuk prediksi yang salah, sedangkan accuracy merupakan nilai Ketika model atau sistem memprediksi dengan benar. Nilai loss adalah angka yang menunjukkan indikasi seberapa buruk prediksi dari model untuk sebuah sampel data. Jika prediksi model sempurna, maka nilai loss akan bernilai nol. Sementara jika yang terjadi sebaliknya, maka jumlah loss semakin banyak. Hal itu juga akan terjadi pada nilai accuracy. Tujuan dari training sebuah model adalah untuk mencari nilai bobot yang memiliki nilai loss rendah dan accuracy yang tinggi.

Gambar 8 menunjukkan grafik accuracy dan loss pada saat proses melatih model dengan menggunakan 30 epoch. Selain menunjukkan proses training juga ditunjukkan proses validation untuk memeriksa proses setiap epoch pada model. Dari hasil training dan validation tidak terdapat perbedaan yang cukup jauh. Hal itu berarti model yang dibangun sudah berlatih dengan baik.



(a)



(b)

Gambar 8. Hasil Grafik Training dan Validation (a) Accuracy dan (b) Loss

Evaluasi dilakukan untuk melihat performa dari model yang telah dibangun. Data validation akan digunakan untuk mengevaluasi model yang sudah di latih. Dari proses ini didapatkan akurasi sebesar 92%.

Classification Report pada *library* sklearn akan digunakan untuk melihat nilai akurasi, presisi, recal dan f1-score. Hasil dari evaluasi dapat dilihat pada

Tabel 2.

Tabel 2. *Classification Report* Model

	precision	recall	F1-score	support
Arbequina	0.93	0.96	0.94	650
Arbosana	0.90	0.91	0.90	509
Changlot	0.90	0.91	0.91	387
Lechin	0.91	0.92	0.92	221
Picual	0.94	0.91	0.93	578
Verdial	0.94	0.88	0.91	328
Accuracy			0.92	2673
Micro avg	0.92	0.92	0.92	2673
Weighted avg	0.92	0.92	0.92	2673

Tabel 3 menunjukkan hasil perbandingan antara model yang diusulkan dengan model yang digunakan pada penelitian sebelumnya[2]. Model yang diusulkan dalam penelitian ini memiliki akurasi yang lebih tinggi dari kedua model sebelumnya. Meskipun begitu ada kategori yang tidak termasuk yaitu *Ocal* karena data yang diperlukan tidak tersedia. Disisi lain, kecepatan dari model yang diusulkan lebih baik dari dua model sebelumnya, terlihat dari model dan ukuran gambar masukkan lebih kecil dari model sebelumnya. Selain itu, waktu yang dibutuhkan untuk melatih model yaitu 164 detik dengan 30 epoch. Sedangkan pada penelitian sebelumnya [2] menggunakan masing-masing 168 dan 74 pada dua model yang berbeda. Perbedaan penggunaan epoch tersebut juga menunjukkan berapa lama model melakukan pelatihan.

Tabel 3. Perbandingan Hasil Model Sebelumnya dengan model yang diusulkan

	AlexNet[2]	Inception-ResnetV2[2]	Propose Model
Arbequina	0.93	0.93	0.96
Arbosana	0.91	0.90	0.91
Changlot	0.86	0.94	0.91
Lechin	0.83	0.88	0.92
Picual	0.94	0.89	0.91
Ocal	0.91	0.93	-
Verdial	0.88	0.93	0.88
Overall	0.89	0.91	0.92
Epoch	168	74	30
Image Size	227 x 227	299 x 299	210 x 210

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, diperoleh kesimpulan bahwa model yang diusulkan mendapatkan akurasi yang hanya lebih baik dari arsitektur model AlexNet dan kurang dari akurasi model InceptionResnetV2. Disisi lain untuk waktu komputasi memiliki kecepatan yang lebih baik dari dua model sebelumnya. Hal itu terlihat dari jumlah epoch yang digunakan dari masing-masing model. Namun pada model yang diusulkan ada satu kategori yang tidak masukkan karena tidak ada ketersediaan data. Oleh karena itu, penelitian selanjutnya yang dapat dilakukan adalah dengan memperbanyak data yang digunakan serta dapat memaksimalkan penggunaan augmentasi data.

DAFTAR PUSTAKA

- [1] A. Cokrowatianto dan J. Wiratmo, "Kajian Potensi Budi Daya Zaitun Ditinjau Dari Kondisi Temperatur Dan Curah Hujan (Studi Kasus: Provinsi NTB Dan NTT)," Bandung: Fakultas Ilmu dan Teknologi Kebumihan Institut Teknologi Bandung; 2016.
- [2] J. M. Ponce, A. Aquino, dan J. M. Andújar, "Olive-Fruit Variety Classification By Means of Image Processing And Convolutional Neural Networks," *IEEE Access*, vol. 7, pp. 147629, 2019.
- [3] A. Kausar, M. Sharif, J. Park, dan D. R. Shin, "Pure-Cnn: A Framework for Fruit Images Classification," *Proc. - 2018 Int. Conf. Comput. Sci. Comput. Intell. Csci 2018*, pp. 404-408, 2018.
- [4] F. F. Maulana dan N. Rochmawati, "Klasifikasi Citra Buah Menggunakan Convolutional Neural Network," *J. Informatics Comput. Sci.*, vol. 01, no. 2, pp. 104-108, 2019.
- [5] S. Ashraf, I. Kadery, A. A. Chowdhury, T. Z. Mahbub, dan R. M. Rahman, "Fruit Image Classification Using Convolutional Neural Networks," *Int. J. Softw. Innov.*, vol. 7, no. 4, pp. 51-70, 2019.
- [6] MM. Ragini, D. Lavanya, B. Likitha, A. Sivaram, dan C. Y. Narayana, "Classification of Fruits And Vegetables With Its Nutrients," *IRJET*, vol:7 no.06, June, 2020.
- [7] S. Mohammed, "Apple Fruits Classification Using Deep Learning," *IJAER*, vol. 3, no. 12, pp. 1-6, 2020.
- [8] Y. D. Zhang, Z. Dong, X. Chen, W. Jia, S. Du, K. Muhammad dan S. H. Wang, "Image Based Fruit Category Classification By 13-Layer Deep Convolutional Neural Network And Data Augmentation," *Multimed. Tools Appl.*, vol. 78, no. 3, pp. 3613-3632, 2019.
- [9] J. L. Rojas-Aranda, J. I. Nunez-Varela, J. C. Cuevas-Tello, dan G. Rangel-Ramirez, "Fruit Classification for Retail Stores Using Deep

- Learning," *Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12088 Incs, pp. 3-13, 2020.
- [10] M. Shamim Hossain, M. Al-Hammadi dan G. Muhammad, "Automatic Fruit Classification Using Deep Learning for Industrial Applications," *IEEE Trans. Ind. Informatics*, vol. 15, no. 2, pp. 1027-1034, 2019.
- [11] M. F. Dzulgarnain, S. Suprpto dan F. Makhrus, "Improvement of Convolutional Neural Network Accuracy on Salak Classification Based Quality on Digital Image," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 13, no. 2, pp. 189-198, 2019.
- [12] A. Asrafil, A. Paliwang, M. R. D. Septian, M. Cahyanti dan R. Swedia, "Klasifikasi Penyakit Tanaman Apel Dari Citra Daun Dengan Convolutional Neural Network," *SEBATIK*, vol.24, no.2, pp. 207-212, 2020.
- [13] Y. Rizki., R. M. Taufiq, H. Mukhtar dan D. Putri, . "Klasifikasi Pola Kain Tenun Melayu Menggunakan Faster R-CNN". *IT Journal Research and Development*, vol. 5, no. 2, pp. 215 - 225, 2021.
- [14] A. Ramdan, V. Zilvan, E. Suryawati, H. F. Pardede dan V. P. Rahadi, "Tea Clone Classification Using Deep CNN With Residual And Densely Connections," *Jurnal Teknologi dan Sistem Komputer.*, vol. 8, no. 4, pp. 289-296, 2020.
- [15] M. Subhan dan H. Basri, "Klasifikasi Mutu Buah Pala (*Myristica Fragrans* Houutt) Berbasis Pengolahan Citra Menggunakan Metode Deep Learning Arsitektur Faster R-Cnn," *INTEK*, vol. 6, no. 2, pp. 106-113, 2019.