

## APLIKASI CONVOLUTIONAL NEURAL NETWORKS (CNN) UNTUK KLASIFIKASI RETAKAN BETON

### APPLICATION OF CONVOLUTIONAL NEURAL NETWORKS (CNN) FOR CLASSIFICATION CRACK OF CONCRETE

Farida Asriani\*<sup>1</sup>, Gandjar Pamudji<sup>2</sup>, Hesti Susilawati<sup>1</sup>, Firdauz Tri Anggoro<sup>1</sup>

\*Email: [farida.asriani@unsoed.ac.id](mailto:farida.asriani@unsoed.ac.id)

<sup>1</sup>Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jenderal Soedirman, Purwokerto

<sup>2</sup>Jurusan Teknik Sipil, Fakultas Teknik, Universitas Jenderal Soedirman, Purwokerto

---

**Abstrak**—Beton menjadi bahan utama dari kebanyakan konstruksi bangunan. Timbulnya sebuah retakan atau kerusakan struktur dari beton tersebut sangat berpengaruh terhadap struktur bangunan secara keseluruhan karena mampu memperpendek umur dari bangunan tersebut. Dari hal tersebut, diperlukannya pengawasan secara rutin terhadap kondisi struktur beton sehingga dapat dilakukan perencanaan pemeliharaan di masa depan. Pada paper ini penulis menerapkan teknologi sistem cerdas terhadap pendeteksian keretakan beton. Penerapan *Deep Learning* dengan arsitektur *Convolutional Neural Networks* dengan model *MobileNet V1* dan *Inception V3* dan *ResNet-50* untuk melakukan pengklasifikasian kondisi keretakan dari sebuah masukan gambar visual. Deteksi keretakan beton yang dilakukan dikelompokkan dalam tiga kelas yaitu retak besar, retak kecil dan tidak retak. Dari hasil *training* dan validasi yang telah dilakukan CNN dengan model *Mobile Net V1* memberikan hasil akurasi yang terbaik yaitu 0,8924 untuk akurasi pelatihan dan 0,8899 untuk akurasi validasi

**Kata kunci** — *Convolutional neural networks*, retakan beton, *Mobile Net V1*, *Inception V3*, *ResNet-50*

---

**Abstract**— *Concrete is the main ingredient of most building constructions. The emergence of a crack or damage to the structure of the concrete is very influential on the overall building structure because it can shorten the life of the building. From this, it is necessary to routinely monitor the condition of concrete structures so that future maintenance plans can be carried out. In this paper the author applies intelligent system technology to detect concrete cracks. Application of Deep Learning with Convolutional Neural Networks architecture with Mobile Net V1 and Inception V3 and ResNet-50 models to classify fracture conditions from an input visual image. The concrete crack detection carried out is grouped into three classes, namely large cracks, small cracks and no cracks. From the results of the training and validation that has been carried out by CNN with the Mobile Net V1 model, it gives the best accuracy results, namely 0.8924 for training accuracy and 0.8899 for validation accuracy*

**Keywords** — *Convolutional neural networks, crack of concrete, Mobile Net V1, Inception V3, ResNet-50*

---

## I. PENDAHULUAN

Retak Beton adalah sebuah fenomena yang sangat sering terjadi pada struktur beton di semua jenis bangunan. Retak beton dapat mengindikasikan ada beberapa perubahan struktur penyusunnya seperti korosi tulangan, perkembangan *spall* atau perubahan kondisi penyangga. Oleh karena itu, pemantauan atau inspeksi pada struktur beton merupakan cara paling efektif untuk mengevaluasi

tingkat keamanan dan untuk mempersiapkan proses perawatan yang sesuai di masa depan [1]-[4].

Metode yang sudah diterapkan untuk menentukan retakan beton adalah dengan menggunakan berbagai instrumen seperti UPV (*Ultrasonic Pulse Velocity*). Perangkat ini dapat mengukur kedalaman dan lebar retakan pada beton dengan mendekatkan perangkat dan memancarkan gelombang ultrasonik dan transduser serta mengukur kecepatan ke gelombang penerima [5]. Selain itu juga telah dilakukan deteksian kerusakan pada beton bertulang

menggunakan fiber optik yang tertanam di dalamnya. Penurunan kualitas beton akibat pembebanan bisa dideteksi dari adanya *losses* pada fiber optik [6],[7],[8].

Telah dikembangkan metode deteksi retak yang efisien dan aman untuk mengatasi berbagai kekurangan deteksi secara manual, di antaranya dengan *image processing* [4]. Dalam beberapa dekade terakhir, komunitas komputer vision telah mengerjakan deteksi otomatis dengan pemrosesan citra di antaranya dengan *thresholding* [9][10], *edge* deteksi [11], transformasi *wavelet* [12][13].

Metode deteksi retakan beton dengan data citra beton yang diolah dengan *Convolutional Neural Network* juga telah diterapkan. Berbagai arsitektur CNN telah dikembangkan untuk tujuan tersebut. Di antaranya CNN dengan arsitektur VGG16 [14]. Arsitektur lain meliputi AlexNet, VGGNet13, dan ResNet18 [15].

Pengembangan aplikasi untuk analisis beton bertulang berbasis android telah dikembangkan sebelumnya [16]. Dengan dikembangkan aplikasi-aplikasi untuk analisis kualitas beton dapat mempermudah manusia dalam melakukan *monitoring* kualitas beton. Pada penelitian ini dilakukan pengembangan aplikasi berbasis CNN untuk klasifikasi keretakan beton. Pada penelitian sebelumnya [14][15] deteksi klasifikasi retakan beton hanya diklasifikasikan pada dua kelas, yaitu retak dan tidak retak. Pada penelitian ini dikembangkan klasifikasi keretakan beton ke dalam tiga kelas, yaitu retak besar, retak kecil dan tidak retak. Adapun arsitektur yang akan diterapkan meliputi MobileNet V1, *Inception V3*, dan ResNet-50. Yang menjadi kontribusi dari penelitian ini adalah pengklasifikasian keretakan beton ke dalam tiga kelas, yaitu tidak retak, retak kecil, dan retak besar. Serta ada perbedaan penggunaan arsitektur CNN yang berbeda, yaitu pada penelitian ini menggunakan CNN dengan arsitektur MobileNet V1, *Inception V3*, dan ResNet 50.

## II. TINJAUAN PUSTAKA

### A. Retakan Beton

Retakan beton adalah kondisi di mana struktur beton terpecah atau terpisah dengan luasan tertentu baik kecil maupun besar yang dapat mengakibatkan kerusakan pada bagian lainnya, sehingga menurunnya daya tekan dari beton tersebut terhadap suatu konstruksi bangunan dan dapat berbahaya bagi lingkungan sekitarnya jika retakan semakin

membesar. Retak pada permukaan beton menjadi salah satu pertanda awal terjadinya degradasi struktur yang sangat penting untuk pemeliharaan bangunan. Retak pada beton dapat menyebabkan masuknya air ke dalam struktur dan menyebabkan korosi pada baja tulangan, dan juga membuat konstruksi memiliki umur lebih pendek. Contoh retakan beton dapat dilihat pada Gambar-1.

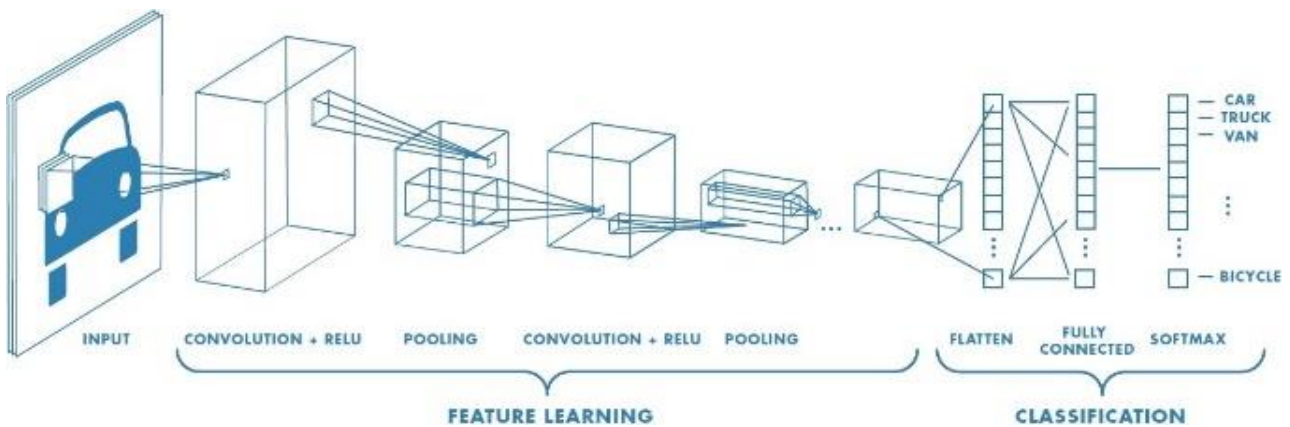


Gambar-1 Contoh Retakan Beton

### B. Convolutional Neural Network

*Convolutional Neural Network* (CNN) adalah salah satu jenis arsitektur *Deep Learning* yang berupa *neural network* yang biasa digunakan untuk mengenali dan mendeteksi objek pada *data image*. CNN terdiri dari neuron memiliki *weight*, *bias* dan *activation function*. Secara garis besar CNN ini menggunakan metode konvolusi dengan menggerakkan sebuah kernel konvolusi (*filter*) dengan ukuran tertentu pada gambar, yang kemudian *computer* mendapatkan informasi representatif baru dari hasil konvolusi pada gambar tersebut.

Arsitektur CNN dibagi menjadi 2 bagian besar yakni *Feature Extraction Layer* dan *Fully-Connected Layer* (MLP), seperti yang terlihat pada Gambar-2.



Gambar-2. Arsitektur CNN.

### C. Arsitektur CNN

#### 1) MobileNet

*MobileNet* merupakan salah satu arsitektur *Convolutional Neural Network* (CNN). Penggunaan lapisan atau *layer* konvolusi pada arsitektur *MobileNets* yakni ketebalan *filter* yang sesuai dengan ketebalan dari *input image*. *MobileNet* membagi konvolusi menjadi *depthwise convolution* dan *pointwise convolution*.

Untuk *MobileNet*, *depthwise convolution* menerapkan satu *filter* ke setiap saluran *input*. *Pointwise convolution* kemudian menerapkan konvolusi 1x1 untuk menggabungkan *output depthwise convolution*. *Standar convolution* memfilter dan menggabungkan *input* menjadi satu set *output* baru dalam satu langkah. *Depthwise convolution* secara mendalam membagi ini menjadi dua lapisan, lapisan terpisah untuk penyaringan dan lapisan terpisah untuk menggabungkan. Faktorisasi ini memiliki efek mengurangi komputasi dan ukuran model secara drastis. Arsitektur *MobileNet* ditunjukkan pada Tabel-1.

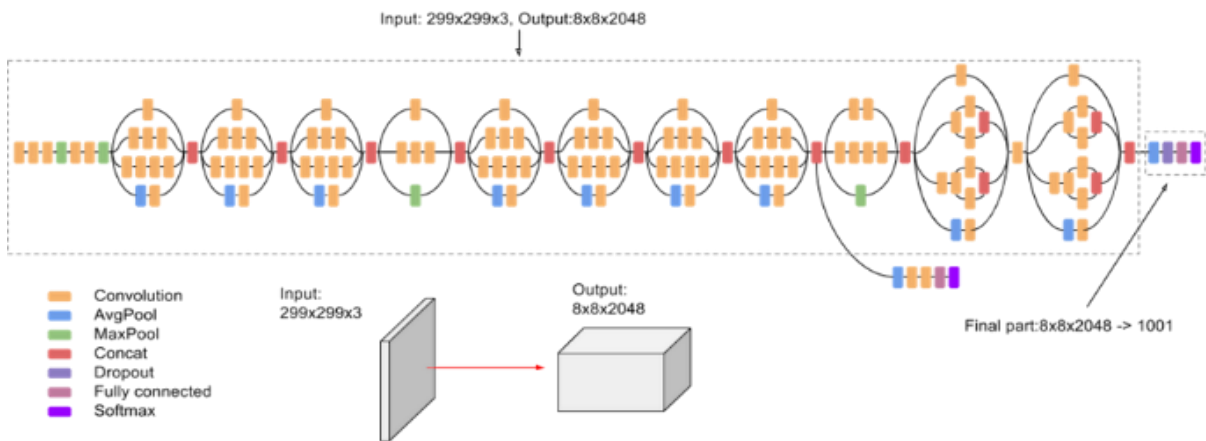
Tabel-1. Arsitektur MobileNet

Type/ Stride	Filter Shape	Input Size
Conv / s2	3 x 3 x 3 x 32	224 x 224 x 3
Conv dw/ s1	3 x 3 x 32 dw	112 x 112 x 32
Conv / s1	1 x 1 x 32 x 64	112 x 112 x 32
Conv dw / s2	3 x 3 x 64 dw	112 x 112 x 64
Conv / s1	1 x 1 x 64 x 128	56 x 56 x 64

Conv dw / s1	3 x 3 x 128 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 128	56 x 56 x 128
Conv dw / s2	3 x 3 x 256 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 256	28 x 28 x 128
Conv dw / s1	3 x 3 x 256 dw	28 x 28 x 128
Conv / s1	1 x 1 x 256 x 256	28 x 28 x 256
Conv dw / s2	3 x 3 x 256 dw	28 x 28 x 256
Conv / s1	1 x 1 x 256 x 512	14 x 14 x 256
5 x Conv dw / s1	3 x 3 x 512 dw	14 x 14 x 512
Conv / s1	1 x 1 x 512 x 512	14 x 14 x 512
Conv dw / s2	3 x 3 x 512 dw	14 x 14 x 512
Conv / s1	1 x 1 x 512 x 1024	7 x 7 x 512
Conv dw / s2	3 x 3 x 1024 dw	7 x 7 x 1024
Conv / s1	1 x 1 x 1024 x 1024	7 x 7 x 1024
Avg Pool / s1	Pool 7 x 7	7 x 7 x 1024
FC/ s1	1024 x 1000	1 x 1 x 1024
Softmax / s1	Classifier	1 x 1 x 1000

#### 2) Inception V3

*Inception V3* merupakan arsitektur yang menjadi juara kedua pada *ImageNet Large-Scale Visual Recognition Challenge 2015* (ILSVRC15) yang merupakan sebuah kompetisi klasifikasi gambar dengan mencapai akurasi 78,1%. Arsitektur dari model ini memiliki 42 *layer* seperti ditunjukkan pada Gambar-3.



Gambar-3. Arsitektur Inception V3.

### 3) ResNet-50

ResNet-50 merupakan bentuk kecil dari *Residual Network* yang menjadi memiliki 48 convolution layer dengan 1 maxpooling layer dan 1 average

layer. *Residual Network* merupakan neural network klasik yang digunakan untuk tugas computer vision. Arsitektur ResNet ditunjukkan pada Tabel-2.

Tabel-2. Arsitektur ResNet.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112x112	7x7, 64, stride 2				
		3x3 max pool, stride 2				
conv2_x	56x56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28x28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14x14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7x7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1x1	average pool, 1000-d fc, softmax				
FLOPs		1.8x10 <sup>9</sup>	3.6x10 <sup>9</sup>	3.8x10 <sup>9</sup>	7.6x10 <sup>9</sup>	11.3x10 <sup>9</sup>

## III. METODE

Penelitian ini dilaksanakan secara eksperimen. Secara garis besar ada empat tahapan penelitian meliputi : 1. pengambilan data dan pre-processing data, 2. Perancangan arsitektur CNN, 3. Pelatihan dan uji validasi CNN, 4. Perbandingan hasil dan pembahasan.

### A. Tahap Pengambilan Data dan Pre-Processing

Dataset beton diunduh dari kaggle yang sudah tersedia gambar dari beton dalam dua folder, yaitu

folder positif dengan isi citra beton dalam keadaan retak dan folder negatif yang berisi citra beton tanpa retakan. Pre-processing dilakukan dengan tujuan untuk memisahkan dataset tersebut ke dalam tiga kelompok, yaitu: retak besar, retak kecil dan tidak retak. Data beton dengan retakan diolah lebih lanjut dengan thresholding untuk mendefinisikan termasuk dalam retakan besar atau retakan kecil.

Jumlah keseluruhan dataset yang diambil sejumlah 25.445 data yang kemudian dikelompokkan menjadi dua bagian, yaitu dataset untuk pelatihan dan dataset untuk pengujian.

## B. Tahap Desain Arsitektur

Pada tahap ini *dataset* akan dilatih menggunakan model arsitektur MobileNet V1 dan InceptionV3 dan ResNet-50. Pada model arsitektur menggunakan MobileNetV1, model dasar ini kemudian diberi *layer* tambahan untuk menambah parameter yang dilatih, *layer* yang ditambahkan yakni *Global Average Pooling 2D*, *Dense 1024 2x* dan *Dense 512 1x* untuk menambahkan parameter yang dilatih dengan aktivasi ReLu, dan Dense 3 dengan aktivasi “softmax” karena hasil nya berupa 3 kelas saja.

Pada model arsitektur menggunakan InceptionV3, model dasar ini kemudian diberi *layer* tambahan untuk menambah parameter yang dilatih, *layer* yang ditambahkan yakni Flatten, Dense 1024 untuk menambahkan parameter yang dilatih dengan aktivasi ReLu, Dropout *layer* dan Dense 3 dengan aktivasi “softmax” karena hasil nya berupa 3 kelas saja.

Pada model arsitektur menggunakan ResNet50, model dasar ini kemudian diberi *layer* tambahan untuk menambah parameter yang dilatih, *layer* yang ditambahkan yakni Flatten, Dense 512 dan Dense 256 untuk menambahkan parameter yang dilatih dengan aktivasi ReLu, dan Dense 3 dengan aktivasi “softmax”.

Pada proses pelatihan *dataset* digunakan fitur *early stopping* sebagai *callbacks*-nya untuk menghentikan pelatihan di saat validasi kesalahannya tidak membesar karena validasi kesalahan akan membesar ketika sudah mencapai puncak terendah validasi kesalahannya.

## C. Tahap Pengujian (Testing)

Pada tahap ini penulis menguji keseluruhan sistem yang telah dirancang. apakah sistem dapat memprediksi keretakan beton dengan tepat. *Dataset* pengujian akan dijadikan bahan uji kepada model yang sudah dilatih untuk melihat keandalan dari model ini. Model akan dievaluasi untuk mengetahui akurasi dan kesalahannya.

Dari hasil pengujian akan dilakukan perhitungan akurasi, dan juga ditentukan arsitektur dengan akurasi terbaiknya

# IV. HASIL DAN PEMBAHASAN

## A. Pengambilan Data dan Pre-Processing

*Dataset* yang digunakan merupakan data citra beton dari kaggle. *Dataset* ini sudah diberi label

*negative* (tidak retak) dan *positive* (retak) dengan jumlah masing-masing 20.000 citra. *Dataset* tersebut dilakukan deteksi luas retakan menggunakan segmentasi dengan metode *Sauvola Thresholding*.

Dari data tersebut peneliti memisahkan data menjadi 2 kelas, yaitu luas retakan 0-5% sebagai retak kecil dan luas retakan >6% sebagai retak besar. Tiap kelas tersebut memiliki jumlah gambar 8.815, kemudian mengambil *dataset negative* berjumlah 8.815 sebagai kelas tidak retak. Data ini yang digunakan sebagai *dataset* pada penelitian ini.

*Dataset* tersebut selanjutnya dikelompokkan lagi menjadi dua bagian, yaitu: *dataset* untuk pelatihan dan *dataset* untuk validasi komposisi seperti pada Tabel-3.

Tabel-3. Pembagian Dataset

Dataset	Persentase	Jumlah Data
Dataset Pelatihan	80%	21.156
Dataset Validasi	16%	4.289

## B. Augmentasi Data

Pada data pelatihan dan validasi dilakukan augmentasi data untuk mendapatkan visual gambar yang lebih banyak dari gambar aslinya. Untuk *dataset* ini gambar akan dibalik secara horizontal dan vertikal, diputar 90°, dan diubah pencahayaannya. Gambar-4 menunjukkan citra asli sebelum proses augmentasi. Gambar-5 menunjukkan citra hasil pembalikan secara horizontal, Gambar-6 menunjukkan hasil pembalikan secara vertikal, Gambar-7 merupakan hasil rotasi citra 90°, dan Gambar-8 menunjukkan hasil citra yang diubah pencahayaannya. Hali ini dilakukan agar model dapat mendeteksi gambar yang memiliki bentuk yang lebih varian dari *dataset*-nya.



Gambar-4. Data citra asli.



Gambar-5. Citra hasil pembalikan secara horizontal



Gambar-6. Citra hasil pembalikan secara vertikal



Gambar-7. Citra hasil rotasi 90°



Gambar-8. Hasil citra yang diubah pencahayaannya

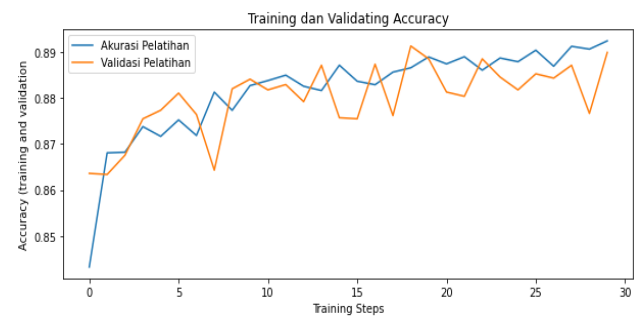
### C. Pelatihan dan Validasi

#### 1) Arsitektur MobileNet V1

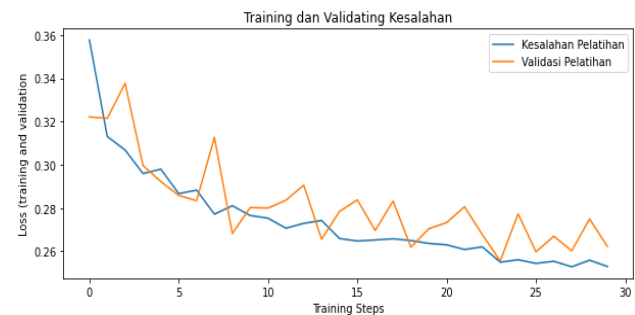
Pada model arsitektur menggunakan MobileNet V1, model dasar ini kemudian diberi *layer* tambahan untuk menambah parameter yang dilatih. *Layer* yang ditambahkan, yaitu *Global Average Pooling 2D*, *Dense 1024 2x* dan *Dense 512 1x* untuk menambahkan parameter yang dilatih dengan aktivasi ReLu, dan *Dense 3* dengan aktivasi

“softmax” karena hasilnya berupa 3 kelas saja. Total parameter yang dilatih adalah 5.854.403 dengan MobileNet V1 berjumlah 3.228.864 parameter dan *layer* tambahan 2.625.539 parameter.

Pada proses pelatihan *dataset* digunakan fitur *early stopping* sebagai *callbacks*-nya untuk menghentikan pelatihan di saat validasi kesalahannya tidak membesar, karena validasi kesalahan akan membesar ketika sudah mencapai puncak terendah validasi kesalahannya.



Gambar-9. Grafik akurasi dan validasi akurasi hasil pelatihan dengan arsitektur MobileNet V1



Gambar-10. Grafik kesalahan dan validasi kesalahan hasil pelatihan dengan arsitektur MobileNet V1

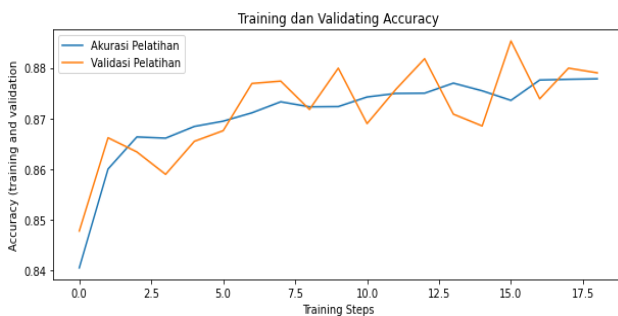
Pada Gambar-9 dan Gambar-10, dapat dilihat kurva hasil pelatihan tiap *epoch*-nya, kurva kuning merupakan kurva validasi dan kurva biru merupakan kurva *training*. Pada awal pelatihan kurva menunjukkan bahwa nilai akurasi terus meningkat dan mengalami penurunan dan kenaikan kembali. Kemudian dilakukan *monitoring* pada validasi kesalahan menggunakan *Early Stopping*. Ketika validasi kesalahan mengalami kenaikan tanpa ada penurunan kembali sampai 6 *epoch* maka pelatihan akan terhenti dan dianggap model sudah *good-fit*. Hal ini digunakan agar model tidak *over-fitting*. Pada pelatihan ini proses terhenti pada *epoch* ke-30 dengan hasil tingkat akurasi 0,8924, validasi akurasi 0,8899, kesalahan 0,2530, dan validasi kesalahan 0,2622. Dapat diartikan bahwa menggunakan

MobileNet V1 mendapatkan hasil yang cukup tinggi pada pelatihan *dataset*.

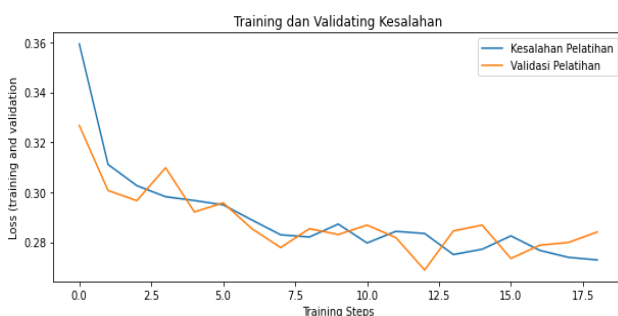
### 2) Arsitektur Inception V3

Pada model arsitektur menggunakan *Inception V3*, model dasar ini kemudian diberi *layer* tambahan untuk menambah parameter yang dilatih, *layer* yang ditambahkan, yaitu *Flatten*, *Dense 1024* untuk menambahkan parameter yang dilatih dengan aktivasi ReLu, *Dropout layer* dan *Dense 3* dengan aktivasi “softmax” karena hasilnya berupa 3 kelas saja. Total parameter yang dilatih adalah 23.904.035 dengan *Inception V3* berjumlah 21.802.784 parameter dan *layer* tambahan 2.101.251 parameter.

Pada proses pelatihan *dataset*, digunakan fitur *early stopping* sebagai *callbacks*-nya untuk menghentikan pelatihan di saat validasi kesalahannya tidak membesar karena validasi kesalahan akan membesar ketika sudah mencapai puncak terendah validasi kesalahannya.



**Gambar-11.** Grafik akurasi dan validasi akurasi hasil pelatihan dengan arsitektur *Inception V3*



**Gambar-12.** Grafik kesalahan dan validasi kesalahan hasil pelatihan dengan arsitektur *Inception V3*

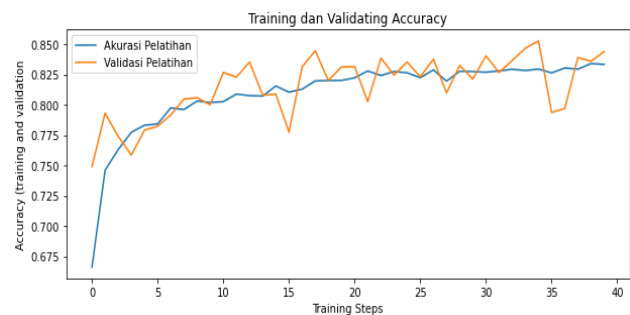
Pada Gambar-11 dan Gambar-12, terlihat kurva hasil pelatihan tiap *epoch*-nya, kurva kuning merupakan kurva validasi dan kurva biru merupakan kurva *training*. Pada awal pelatihan kurva menunjukkan bahwa nilai akurasi terus meningkat dan mengalami penurunan dan kenaikan kembali. Kemudian dilakukan *monitoring* pada validasi

kesalahan menggunakan *Early Stopping*. Ketika validasi kesalahan mengalami kenaikan tanpa ada penurunan kembali sampai 6 *epoch* maka pelatihan akan berhenti dan dianggap model sudah *good-fit*. Hal ini digunakan agar model tidak *over-fitting*. Pada pelatihan ini proses berhenti pada *epoch* ke-19 dengan hasil tingkat akurasi 0,8778, validasi akurasi 0,8789, kesalahan 0,2729, dan validasi kesalahan 0,2841. Dapat diartikan bahwa menggunakan *Inception V3* mendapatkan hasil yang cukup tinggi pada pelatihan *dataset*.

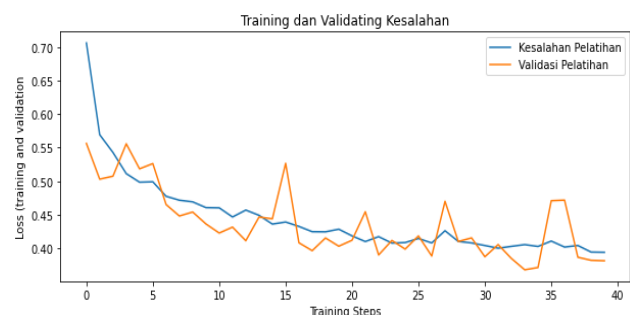
### 3) Arsitektur ResNet 50

Pada model arsitektur menggunakan ResNet50, model dasar ini kemudian diberi *layer* tambahan untuk menambah parameter yang dilatih, *layer* yang ditambahkan, yaitu *Flatten*, *Dense 512* dan *Dense 256* untuk menambahkan parameter yang dilatih dengan aktivasi ReLu, dan *Dense 3* dengan aktivasi “softmax” karena hasilnya berupa 3 kelas saja. Total parameter yang dilatih adalah 27.912.627 dengan ResNet50 berjumlah 23.587.712 parameter dan *layer* tambahan 4.326.915 parameter.

Pada proses pelatihan *dataset* digunakan fitur *early stopping* sebagai *callbacks*-nya untuk menghentikan pelatihan disaat validasi kesalahannya tidak membesar karena validasi kesalahan akan membesar ketika sudah mencapai puncak terendah validasi kesalahannya.



**Gambar-13.** Grafik akurasi dan validasi akurasi hasil pelatihan dengan arsitektur ResNet 50

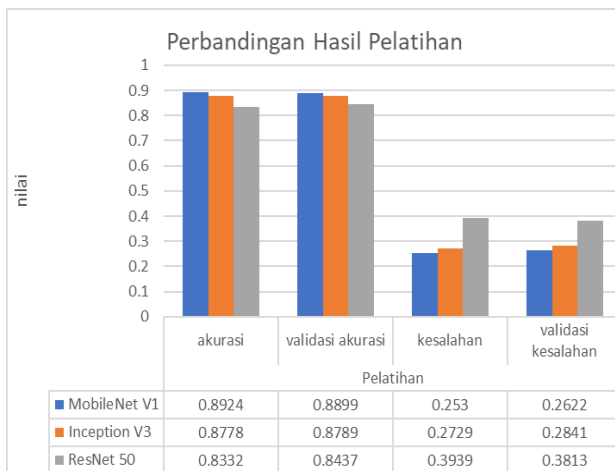


**Gambar-14.** Grafik kesalahan dan validasi kesalahan hasil pelatihan dengan arsitektur ResNet 50

Pada Gambar-13 dan Gambar-14, terlihat kurva hasil pelatihan tiap *epoch*-nya, kurva kuning merupakan kurva validasi dan kurva biru merupakan kurva *training*. Pada awal pelatihan kurva menunjukkan bahwa nilai akurasi terus meningkat dan mengalami penurunan dan kenaikan kembali. Kemudian dilakukan *monitoring* pada validasi kesalahan menggunakan *Early Stopping*. Ketika validasi kesalahan mengalami kenaikan tanpa ada penurunan kembali sampai 6 *epoch* maka pelatihan akan berhenti dan dianggap model sudah *good-fit*. Hal ini digunakan agar model tidak *over-fitting*. Pada pelatihan ini proses berhenti pada *epoch* ke-40 dengan hasil tingkat akurasi 0,8332, validasi akurasi 0,8437, kesalahan 0,3939, dan validasi kesalahan 0,3813. Dapat diartikan bahwa menggunakan ResNet 50 mendapatkan hasil yang cukup tinggi pada pelatihan *dataset*.

#### D. Perbandingan Hasil Training dan Validasi

Berdasarkan percobaan ketiga arsitektur di atas dapat dibandingkan keandalannya untuk mengetahui arsitektur yang terbaik untuk melakukan deteksi retak beton.



Gambar-15. Perbandingan Hasil Pelatihan

Hasil pelatihan dan validasi CNN dengan arsitektur MobileNet V1, *Inception V3* dan ResNet 50 ditampilkan dalam tabulasi pada Gambar-15. MobileNet V1 memiliki akurasi pelatihan dan akurasi validasi tertinggi dibandingkan dengan *Inception V3* dan ResNet 50. Ditinjau dari kesalahan pelatihan dan kesalahan validasinya juga MobileNet memiliki kesalahan pelatihan dan kesalahan validasi terendah dibandingkan dengan *Inception V3* maupun ResNet 50. Dapat disimpulkan bahwa pada penelitian ini, arsitektur terbaik dapat digunakan

untuk deteksi retak beton berbasis CNN adalah menggunakan MobileNet V1 dengan akurasi pelatihan 0,8924 dan akurasi validasi mencapai 0,8899.

## V. PENUTUP

### A. Kesimpulan

Pada *paper*, klasifikasi retakan beton ke dalam tiga kelas retakan, yaitu: retak besar, retak kecil dan tidak retak. Ada tiga arsitektur CNN yang digunakan meliputi: MobileNet V1, *Inception V3* dan ResNet 50. Berdasarkan hasil *training* dan validasi diperoleh hasil arsitektur terbaik untuk deteksi retak beton adalah dengan menggunakan MobileNet V1. Dengan akurasi pelatihan 0,8924 dan akurasi validasi mencapai 0,8899.

### B. Saran

Saran untuk penelitian selanjutnya bisa dikembangkan klasifikasi dengan arsitektur lain dan juga pengembangan ke sistem identifikasi luas dan kedalaman retakan beton.

## UCAPAN TERIMA KASIH

Terimakasih yang sebesar-besarnya tim kami persembahkan untuk Universitas Jenderal Soedirman melalui Lembaga Penelitian dan Pengabdian Kepada Masyarakat yang telah memberikan dukungan pendanaan untuk terlaksanakannya penelitian ini.

## DAFTAR PUSTAKA

- [1] S. Dorafshan, M. Maguire, and X. Qi, "Automatic Surface Crack Detection in Concrete Structures using Otsu Thresholding and Morphological Operations", Utah State University. April 2016.
- [2] D. G. Aggelis, N. Alver, and H. K. Chai. Health monitoring of civil infrastructure and materials," *Scientific World Journal*, vol. 2014, 2014.
- [3] I.-H. Kim, H. Jeon, S.-C. Baek, W.-H. Hong, and H.-J. Jung, "Application of Crack Identification Techniques for an Aging Concrete Bridge Inspection Using an Unmanned Aerial Vehicle", *Sensors* 18, no. 6: 1881, 2018.
- [4] T. Liu, H. Huang, and Y. Yang, "Crack Detection Of Reinforced Concrete Member Using Rayleigh-Based Distributed Optic Fiber Strain Sensing System", *Advances in Civil Engineering*, vol. 2020, 2020.
- [5] F. S. Herlambang and E. Y. Setyono, "Pengaruh Jarak Transduser dan Tulangan pada Pengukuran



- Kedalaman Retak Beton Menggunakan Ultrasonic Pulse Velocity (UPV),”* Prosiding Sentrinov, vol. 3, No. 1, pp. 162–172, 2017.
- [6] F. Asriani, G. Pamudji, H. Susilawati, and S. W. Ismani, “Damage Detection Tool Design of Lightweight Concrete Using Optical Fiber Sensor and Phototransistor,” *Ijitee*, vol. 1, no. 1, pp. 8–12, 2017.
- [7] Asriani F, Pamudji G. and Susilawati H., “*Fiber-optic as embedded sensors to failure detection of beam green concrete Fiber Optic as Embedded Sensors to Failure Detection of Beam Green Concrete,*” AIP Conf. Proc., vol. 020016, issue.1, 2019.
- [8] Asriani F, Winasis and Pamudji G, “*Sensitivity of Optical Fiber Sensors to Deflection of Reinforced Concrete Beam*”, IOP Conference Series: Materials Science and Engineering, ICETIR 2020
- [9] Y.-C. Tsai, V. Kaul, and R. M. Mersereau, “Critical assessment of pavement distress segmentation methods,” *Journal of Transportation Engineering*, vol. 136, no. 1, pp. 11–19, 2010.
- [10] D. Zhang, Q. Li, Y. Chen, M. Cao, L. He, and B. Zhang, “An efficient and reliable coarse-to-fine approach for asphalt pavement crack detection,” *Image and Vision Computing*, vol. 57, pp. 130–146, 2017.
- [11] A. Ayenu-Prah and N. Atttoh-Okine, “Evaluating Pavement Cracks with Bidimensional Empirical Mode Decomposition,” *EURASIP Journal on Advances in Signal Processing*, vol. 2008, pp. 1–7, 2008.
- [12] P. Subirats, J. Dumoulin, V. Legeay, and D. Barba, “Automation of pavement surface crack detection using the continuous wavelet transform,” in Proceedings of the International Conference on Image Processing, pp. 3037–3040, IEEE, Atlanta, GA, USA, 2006.
- [13] L. Ying and E. Salari, “Beamlet Transform-Based Technique for Pavement Crack Detection And Classification,” *Computer Aided Civil and Infrastructure Engineering*, vol. 25, no. 8, pp. 572–580, 2010.
- [14] Wilson Ricardo Leal da Silva and Diogo Schwerz de Lucena, “Concrete Cracks Detection Based on Deep Learning Image Classification”, Proceedings 2, no.8: 489; doi:10.3390/ICEM18-05387. 2018.
- [15] Cheng Yang, Jingjie Chen Zhiyuan Li and Yi Huang, “Structural Crack Detection and Recognition Based on Deep Learning, Applied Sciences, vol. 11, no. 6, p. 2868, 2021.
- [16] Prio Handoko and Agustinus Agus Setiawan, “Pengembangan Aplikasi Analisis Penampang Beton Bertulang berbasis Android”, *Dinamika Rekayasa*, vol.13, no.2, pp. 69-80. 2017.