

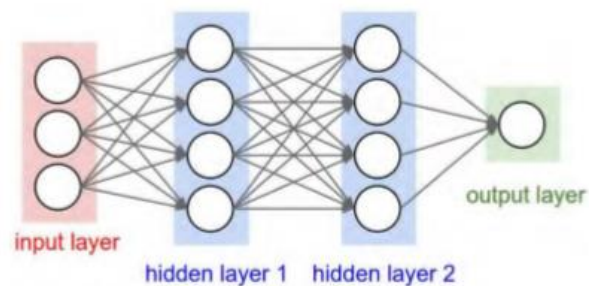
## BAB II

### TINJAUAN PUSTAKA

#### 2.1 *Convolutional Neural Network (CNN)*

CNN adalah variasi dari *Multilayer Perceptron (MLP)* yang didesain untuk mengolah data dua dimensi. CNN juga merupakan jenis dari *deep neural network* yang didesain untuk mengolah data dua dimensi dengan kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra (Setyanto, dkk., 2020). CNN pertama kali dikembangkan dengan nama NeoCognitron oleh Kunihiko Fukushima, seorang peneliti dari NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Jepang (Fukushima, 1980). Konsep tersebut kemudian dimatangkan oleh Yann LeCun, seorang peneliti dari AT&T Bell Laboratories di Holmdel, New Jersey, USA. Model CNN dengan nama LeNet berhasil diterapkan oleh LeCun pada penelitiannya mengenai pengenalan angka dan tulisan tangan (LeCun, dkk., 1998).

##### 2.1.1 Konsep *Convolutional Neural Network*



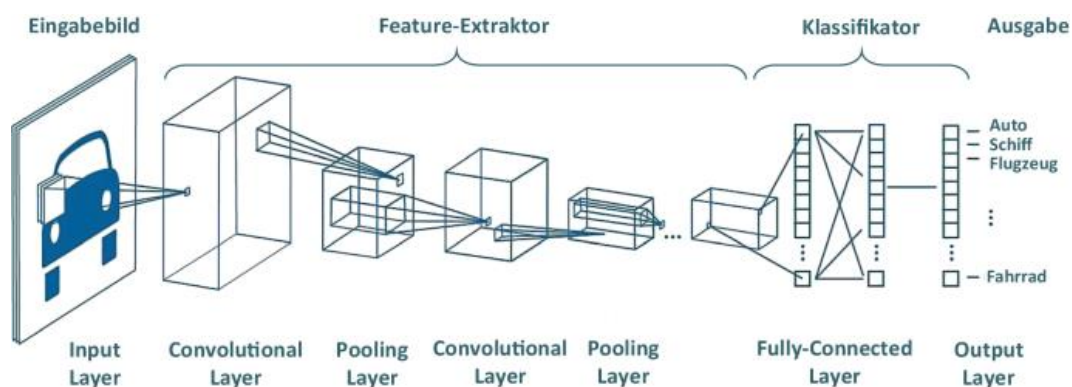
Gambar II.1 Arsitektur Sederhana *Multilayer Perceptron* (Yuliani, dkk., 2019)

Sebuah MLP seperti pada Gambar II.1 memiliki  $i$  *layer* (kotak merah dan biru) dengan masing-masing *layer* berisi  $j$  *neuron* (lingkaran putih). MLP menerima input data satu dimensi dan mempropagasikan data tersebut pada jaringan hingga menghasilkan *output*. Setiap hubungan antar *neuron* pada dua *layer* yang bersebelahan memiliki parameter bobot satu dimensi yang menentukan kualitas mode. Di setiap data input pada *layer* dilakukan operasi *linear* dengan nilai bobot yang ada, kemudian hasil komputasi akan ditransformasi menggunakan operasi *non-linear* yang disebut sebagai fungsi aktivasi. Pada CNN, data yang dipropagasikan oleh jaringan adalah data dua dimensi, sehingga operasi *linear* dan parameter bobot pada CNN berbeda. Pada CNN operasi *linear* menggunakan operasi konvolusi dan bobot berbentuk empat dimensi yang merupakan kumpulan kernel konvolusi.

### 2.1.2 Arsitektur *Convolutional Neural Network*

*Neural networks* terdiri dari berbagai *layer* dan di setiap *layer* terdapat beberapa *neuron* (Stathakis, 2008). MLP memiliki keterbatasan untuk memetakan persamaan dengan data yang lebih kompleks. Pada permasalahan jumlah *hidden layer* dibawah tiga *layer*, terdapat pendekatan untuk menentukan jumlah *neuron* pada masing-masing *layer* untuk mendekati hasil optimal. Dengan berkembangnya *deep learning*, ditemukan bahwa untuk mengatasi kekurangan MLP dalam menangani data kompleks, diperlukan fungsi untuk mentransformasi data input menjadi bentuk yang lebih mudah dimengerti oleh MLP. Hal tersebut memicu berkembangnya model *neural network* dengan jumlah *layer* diatas tiga. CNN

merupakan suatu *layer* yang memiliki susunan *neuron* 3D (lebar, tinggi, kedalaman). Lebar dan tinggi merupakan ukuran *layer* sedangkan kedalaman mengacu pada jumlah *layer*. Dengan demikian, CNN merupakan metode untuk mentransformasikan gambar original *layer by layer* dari nilai *pixel* gambar kedalam nilai skoring kelas untuk klasifikasi. Setiap *layer* ada yang memiliki *hyperparameter* dan ada yang tidak (bobot dan bias pada *neuron*) (Zufar & Setiyono, 2016). LeNet-5 merupakan sebuah jaringan konvolusional 7-level yang awalnya dikembangkan oleh LeCun pada tahun 1998 (Haridas, dkk., 2019). Arsitektur LeNet-5 digunakan untuk mengekstraksi fitur antarmuka BrainComputer Sakhavi, dkk., (2018) dengan mempertimbangkan *common spatial patterns* (CSPs) *algorithm*. Tetapi masalah utama yang terjadi ialah akurasi rata-rata yang diperoleh oleh jaringan yang dimodifikasi lebih rendah dari akurasi aslinya.

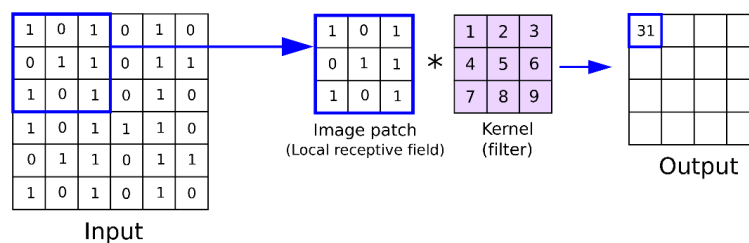


Gambar II.2 Arsitektur *Convolutional Neural Network* (Zschech, dkk., 2021)

Berdasarkan Gambar II.2, arsitektur CNN terbagi menjadi dua bagian besar yaitu: *convolutional layer* (terdiri dari *pooling layer*, *activation function* dan *hyperparameter*) dan *fully connected layer*.

a. *Convolution Layer*

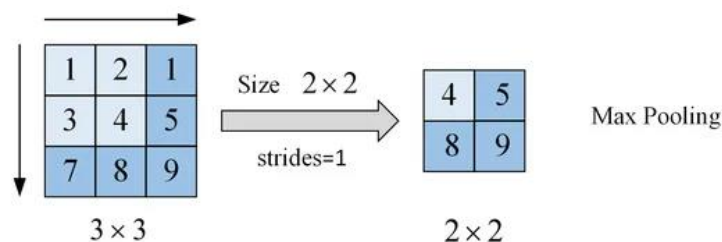
*Convolution layer* merupakan bagian dari tahap pada arsitektur CNN. Tahap ini melakukan operasi konvolusi pada *output* dari *layer* sebelumnya. *Layer* tersebut adalah proses utama yang mendasari jaringan arsitektur CNN. Konvolusi adalah istilah matematis dimana pengaplikasian sebuah fungsi pada *output* fungsi lain secara berulang. Operasi konvolusi merupakan operasi pada dua fungsi argumen bernilai nyata. Operasi ini menerapkan fungsi *output* sebagai *feature map* dari *input* citra. *Input* dan *output* ini dapat dilihat sebagai dua argumen bernilai riil. *Convolutional Layer* terdiri dari neuron yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (*pixels*). Sebagai contoh, *layer* pertama pada *feature extraction layer* biasanya adalah *conv. Layer* dengan ukuran 5x5x3. Panjang 5 *pixels*, tinggi 5 *pixels* dan tebal/jumlah 3 buah sesuai dengan channel dari *image* tersebut. Ketiga filter ini akan digeser keseluruhan bagian dari gambar. Setiap pergeseran akan dilakukan operasi “dot” antara *input* dan nilai dari filter tersebut sehingga menghasilkan sebuah *output* atau biasa disebut sebagai activation map atau *feature map*. Perhatikan ilustrasi berikut :



Gambar II.3 Ilustrasi *Feature Map* (Nash, dkk., 2015)

#### b. *Pooling*

*Pooling* merupakan pengurangan ukuran matriks dengan menggunakan operasi *pooling*. *Pooling layer* biasanya berada setelah lapisan konvolusi. Pada dasarnya *pooling layer* terdiri dari sebuah filter dengan ukuran dan stride tertentu yang akan secara bergantian bergeser pada seluruh area *feature map*. Dalam *pooling layer* terdapat dua macam *pooling* yang biasa digunakan yaitu *average pooling* dan *max-pooling*. Nilai yang diambil pada *average pooling* adalah nilai rata-rata, sedangkan pada *max-pooling* adalah nilai maksimal. Lapisan *Pooling* yang dimasukkan diantara lapisan konvolusi secara berturut-turut dalam arsitektur model CNN dapat secara progresif mengurangi ukuran volume *output* pada *feature map*, sehingga mengurangi jumlah parameter dan perhitungan di jaringan, untuk mengendalikan *overfitting*. Lapisan *pooling* bekerja di setiap tumpukan *feature map* dan melakukan pengurangan pada ukurannya. Bentuk lapisan *pooling* umumnya dengan menggunakan filter dengan ukuran 2x2 yang diaplikasikan dengan langkah sebanyak dua dan beroperasi pada setiap irisan dari *inputnya*. Berikut ini adalah contoh gambar operasi *max-pooling* :

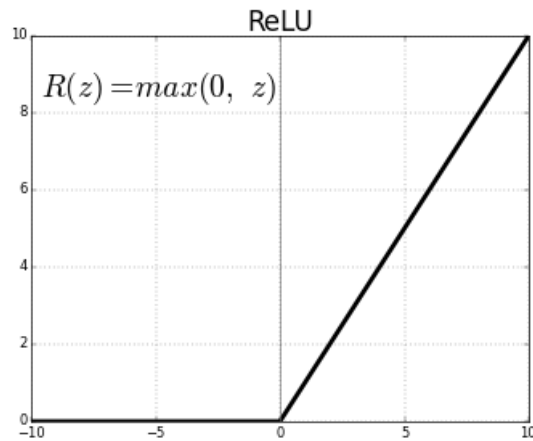


Gambar II.4 Contoh Operasi *MaxPooling* (Chen, dkk., 2021)

Gambar II.4 menunjukkan proses dari *max-pooling*. *Output* dari proses pooling adalah sebuah matriks dengan dimensi yang lebih kecil dibandingkan dengan citra awal. Lapisan pooling diatas akan beroperasi pada setiap irisan kedalaman volume *input* secara bergantian. Jika dilihat dari gambar diatas operasi *max-pooling* dengan menggunakan ukuran filter 2x2. Masukan pada proses tersebut berukuran 3x3, dari masing-masing 3 angka pada *input* operasi tersebut diambil nilai maksimalnya kemudian dilanjutkan membuat ukuran *output* baru menjadi ukuran 2x2.

c. *Activation Function : ReLu (Rectified Linear Units)*

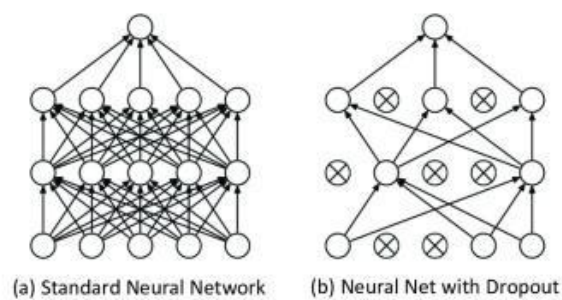
*Rectified Linear Units (ReLU)* layer ini mengaplikasikan fungsi  $f(x) = \max(0, x)$  untuk meningkatkan sifat nonlinearitas fungsi keputusan dan jaringan secara keseluruhan tanpa mempengaruhi bidang-bidang reseptif pada *convolutional layer*. Contoh operasi ReLU seperti pada Gambar II.5.



Gambar II.5 Ilustrasi *Feature Map* (Jyothula, 2020)

d. *Dropout*

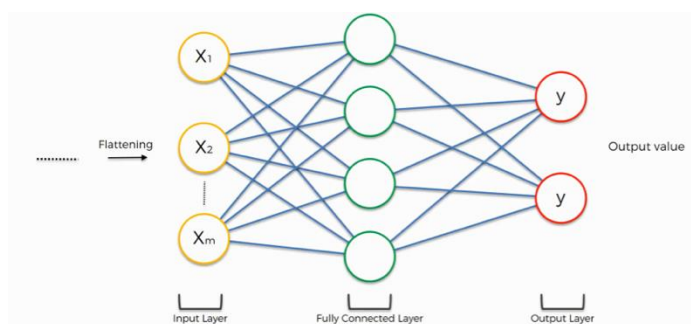
Dropout merupakan sebuah teknik regulasi jaringan syaraf dengan tujuan memilih beberapa *neuron* secara acak dan tidak akan dipakai selama proses pelatihan, dengan kata lain *neuron-neuron* tersebut dibuang secara acak. Hal ini berarti bahwa kontribusi neuron yang dibuang akan diberhentikan sementara jaringan dan bobot baru juga tidak diterapkan pada *neuron* pada saat melakukan *backpropagation*. Gambar II.6 merupakan contoh proses dari dropout.



Gambar II.6 Ilustrasi *Dropout* (Fawwaz, dkk., 2021)

e. *Fully Connected Layer*

*Fully-Connected Layer* adalah sebuah lapisan dimana semua neuron aktivasi dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya sama seperti halnya dengan neural network biasa. Pada dasarnya lapisan ini biasanya digunakan pada MLP (*Multi Layer Perceptron*) yang mempunyai tujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linier. Perbedaan antara lapisan *fully-connected* dan lapisan konvolusi biasa adalah neuron di lapisan konvolusi terhubung hanya ke daerah tertentu pada *input*, sementara lapisan *fully-connected* memiliki neuron yang secara keseluruhan terhubung. Namun, kedua lapisan tersebut masih mengoperasikan produk dot, sehingga fungsinya tidak begitu berbeda. Gambar II.7 adalah contoh dari proses *fully-connected*.



Gambar II.7 Proses *Fully Connected Layer* (Santosa & Ariyanto, 2018)

f. *Softmax Classifier*

*Softmax Classifier* adalah generalisasi dari fungsi logistik. *Output* dari softmax ini dapat digunakan untuk mewakili distribusi sebuah kategori. *Softmax function* digunakan dalam berbagai macam metode klasifikasi



contohnya *multinomial logistic regression*, *multiclass linear discriminant analysis*, *naive Bayes classifier*, dan *neural network*. Secara spesifiknya fungsi ini biasa digunakan pada metode klasifikasi *multinomial logistic regression* dan *multiclass linear discriminant analysis*. Fungsi yang digunakan dapat dilihat pada persamaan (1).

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \quad (1)$$

Notasi  $f_j$  menunjukkan hasil fungsi untuk setiap elemen ke- $j$  pada vektor keluaran kelas. Argumen  $z$  adalah hipotesis yang diberikan oleh model pelatihan agar dapat diklasifikasi oleh fungsi softmax. Softmax juga memberikan hasil yang lebih intuitif dan juga memiliki interpretasi probabilistik yang lebih baik dibanding algoritma klasifikasi lainnya. Softmax memungkinkan kita untuk menghitung probabilitas untuk semua label. Dari label yang ada akan diambil sebuah vektor nilai bernilai riil dan merubahnya menjadi vektor dengan nilai antara nol dan satu yang bila semua dijumlah akan bernilai satu.

## 2.2 Adaptive Kernel

*Adaptive kernel* merupakan sebuah modul yang memiliki kemampuan untuk secara otomatis melakukan penyesuaian terhadap kebutuhan. Tujuan utama dari modul ini adalah untuk memodelkan sistem secara dinamis. Permasalahan utama yang atasi adalah terkait variabel-variabel dengan sifat yang bervariasi, memiliki nilai berbeda-beda, dan tidak tetap. Modul ini memiliki dua fungsi, yaitu *utility*

*function* berupa fungsi untuk mengurutkan atau mendefinisikan ragam alternatif menurut kegunaannya, dan *learner function* yaitu melakukan pembelajaran dan pengenalan untuk memperoleh hasil yang paling optimal. Fungsi kernel didapatkan dengan menggunakan hasil dari konvolusi tiap input konvolusi  $Q_{u,v}$  seperti pada persamaan (Zamora, dkk., 2019):

$$\sigma_{u,v} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} Q_{(u,v)_{i,j}} x_{i,j} \quad (2)$$

Kernel baru  $K_{u,v}$  dapat digunakan selanjutnya untuk melakukan konvolusi pada input gambar untuk menghasilkan  $S$  yang selanjutnya dijadikan sebagai kernel *output* seperti pada perhitungan (Zamora, dkk., 2019):

$$S = \sum_{u,v} x_{u,v} K \left( \sum_{i,j} Q_{u,v,i,j} x_{i,j} \right) \quad (3)$$

### 2.3 Adagrad

Adagrad adalah varian dari *Stochastic Gradient Descent* yang mengubah *learning rate* pada setiap parameter dalam proses pelatihan, oleh karena itu disebut *Adaptive Gradient Descent Algorithm* (Zhang, dkk., 2018). Pada tahun 2011, Adagrad diperkenalkan oleh (Duchi, dkk., 2011). Adagrad memodifikasi tingkat pembelajaran yang beradaptasi dengan arah penurunan menuju nilai optimum. Dengan kata lain, Adagrad mempertahankan tingkat pembelajaran yang rendah untuk fitur yang sering muncul dan tingkat pembelajaran yang tinggi untuk fitur yang jarang muncul (Pennington, dkk., 2014). Kemudian, Adagrad menerapkan tingkat pembelajaran yang berbeda untuk setiap parameter pada setiap waktunya (Lydia & Francis, 2019).

## 2.4 Studi Literatur Penelitian

Literatur yang dipakai sebagai referensi dalam penelitian ini adalah jurnal-jurnal ilmiah nasional dan internasional serta modul mengenai *machine learning*. Literatur-literatur tersebut akan menjadi pedoman untuk memudahkan penelitian.

Beberapa tahun terakhir *Convolutional Neural Networks* telah sangat sukses dalam pengenalan gambar dan masalah klasifikasi gambar. Beragam metode telah digunakan dalam mengembangkan CNN oleh para peneliti. Sebagai contoh penelitian yang dilakukan oleh (Kabir, dkk., 2020). Penelitian tersebut menggunakan struktur SpinalNet yang merupakan penggabungan dari implementasi CNN dan Gradual Input. Penelitian berfokus pada kekurangan dari CNN terkait intensitas komputasi yang diakibatkan oleh besarnya jaringan yang diinputkan. Maka dari itu, *gradual input* diterapkan pada penelitian ini yang merupakan konsep inputan secara bertahap sehingga mengurangi beban proses perhitungan. Hasil penelitian menunjukkan bahwa SpinalNet mampu meningkatkan akurasi dari model CNN biasanya. Penelitian lain dilakukan oleh Ali, dkk., (2020), penelitian tersebut mengintegrasikan antara model CNN dengan algoritma Extreme Learning Machine (ELM). ELM berfungsi untuk mengklasifikasikan gambar dan memiliki kelebihan yaitu kecepatan dalam proses pembelajaran. Hasil penelitian menunjukkan bahwa model tersebut memiliki akurasi lebih baik dibandingkan dengan model CNN konvensional. Penelitian (Ayumi, dkk., 2016) mengembangkan CNN dengan mengoptimalkannya menggunakan algoritma *metaheuristic* yaitu algoritma *Microcanonical Annealing* yang merupakan varian dari *Simulated Annealing*. Penelitian tersebut mencoba untuk meningkatkan

kecepatan komputasi dari proses pembelajaran model. Hasil menunjukkan bahwa model penelitian dapat meningkatkan proses pembelajaran di setiap *epoch*-nya sampai 4,60%. Penelitian Kadam, dkk., (2020) menggunakan lima arsitektur berbeda dengan berbagai *convolution layer*, *filter size*, dan *fully connected layer*. Eksperimen dilakukan dengan hyperparameter yang bervariasi yaitu fungsi aktivasi, *optimizer*, *learning rate*, *dropout rate* dan *batch size*. Hasil penelitian menunjukkan bahwa pemilihan fungsi aktivasi, *optimizer* dan *dropout rate* berdampak pada hasil akurasi. Semua arsitektur yang digunakan dalam penelitian memberikan hasil akurasi lebih dari 99%. Biswas & Islam (2021) penelitian ini memfokuskan pada dua batasan masalah CNN yaitu akurasi rendah dan kecepatan komputasi yang lambat. Penelitian mencoba memodelkan arsitektur CNN yang efisien guna memecahkan permasalahan yang ada. Terdapat lima arsitektur CNN yang dimodelkan untuk diterapkan pada dua *dataset* yang berbeda. Hasil penelitian menunjukkan bahwa dari lima arsitektur yang ada, arsitektur yang dilengkapi *stochastic gradient descent* dengan *momentum optimizer* menunjukkan hasil yang terbaik.

Penggunaan literatur secara rinci dapat dilihat pada Tabel II.1 yang merupakan penelitian terkait penggunaan model CNN dari penelitian sebelumnya.

**Tabel II.1 Penelitian Terkait Penggunaan Model CNN**

No	Peneliti	Judul	Hasil Penelitian
1	(Prilianti, dkk., 2019)	Performance Comparison of the Convolutional Neural Network Optimizer for Photosynthetic Pigments Prediction on Plant Digital Image	Penelitian bertujuan untuk meningkatkan akurasi model CNN dengan melakukan optimalisasi pada pemilihan parameter arsitektur CNN yaitu metode optimasi dan <i>loss function</i> . Penelitian menggunakan tujuh algoritma <i>stochastic gradient descent</i> Hasil eksperimen penelitian ini menghasilkan bahwa dari ke tujuh algoritma optimasi yang digunakan, Adam

			merupakan algoritma dengan hasil terbaik yang digunakan pada <i>architecture</i> LeNet untuk memprediksi <i>pigment plant digital image</i> .
https://aip.scitation.org/doi/10.1063/1.5094284			AIP Conference Proceedings
2	(Syulistyo, dkk., 2016)	Particle Swarm Optimization (PSO) for Training Optimization on Convolutional Neural Network (CNN)	Penelitian ini berfokus pada kekurangan neural network yaitu apabila diterapkan pada aplikasi yang kompleks maka akurasi menurun secara signifikan. Maka dari itu, penelitian ini memanfaatkan penggunaan PSO pada proses pelatihan yang bertujuan untuk mengoptimalkan hasil vektor solusi pada CNN guna meningkatkan akurasi pengenalan. Hasil eksperimen menunjukkan bahwa penggunaan PSO pada CNN dapat meningkatkan tingkat akurasi. Dari empat <i>epoch</i> yang dilakukan CNNPSO dapat mencapai akurasi 95.08%.
https://jiki.cs.ui.ac.id/index.php/jiki/article/view/366			Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
3	(Dubey, dkk., 2020)	diffGrad: An Optimization Method for Convolutional Neural Networks	Penelitian menggunakan diffGrad sebagai algoritma untuk mengoptimalkan penggunaan CNN. DiffGrad merupakan sebuah perbedaan antara gradien masa kini dan masa lalu. Ukuran langkah disesuaikan untuk setiap parameter sedemikian rupa sehingga harus memiliki ukuran langkah yang lebih besar untuk parameter perubahan gradien yang lebih cepat dan ukuran langkah yang lebih rendah untuk parameter perubahan gradien yang lebih rendah. Hasil eksperimen menunjukkan bahwa diffGrad memiliki tingkat akurasi lebih tinggi daripada algoritma SGD (SGDM, Adagrad, RMSProp, AMSGrad dan Adam). Penelitian ini juga menggunakan beberapa fungsi aktivasi CNN yang berbeda.
https://ieeexplore.ieee.org/document/8939562			IEEE Transactions on Neural Networks and Learning Systems
4	(Lee, dkk., 2021)	Genetic Algorithm Based Deep Learning Neural Network Structure and Hyperparameter Optimization	Eksperimen pada penelitian ini menggunakan Algoritma <i>genetic</i> sebagai Algoritma optimasi untuk CNN. Hasil dari penelitian ini menunjukkan bahwa algoritma genetika dengan jumlah percobaan 50 generation yang diujikan pada dataset penyakit Alzemaier menunjukkan tingkat akurasi yang lebih tinggi dengan tingkat akurasi 81.74%.
https://www.mdpi.com/2076-3417/11/2/744			MDPI Applied Sciences
5	(Andika, dkk., 2019)	Klasifikasi Penyakit Pneumonia Menggunakan Metode	Hasil eksperimen menunjukkan bahwa penerapan CNN dengan optimasi Algoritma Adam pada dataset labeled Optical Coherence Tomography dan Chest X-Ray Image for Classification menghasilkan akurasi

		Convolutional Neural Network dengan Optimasi Adaptive Momentum	98,98% pada data training dan 97,00% pada data validasi dengan menggunakan 100 epoch.
<a href="https://journal.stats.id/index.php/ijsa/article/view/560">https://journal.stats.id/index.php/ijsa/article/view/560</a>			Indonesian Journal of Statistics and Its Applications
6	(Kabir, dkk., 2020)	SpinalNet: Deep Neural Network with Gradual Input	Penelitian ini menerapkan arsitektur SpinalNet pada CNN. Penelitian berfokus pada kekurangan dari CNN terkait intensitas komputasi yang diakibatkan oleh besarnya jaringan yang diinputkan. Hasil penelitian menunjukkan bahwa SpinalNet mampu meningkatkan akurasi dari model CNN biasanya
<a href="https://arxiv.org/abs/2007.03347">https://arxiv.org/abs/2007.03347</a>			Computer Vision and Pattern Recognition
7	(Biswas & Islam, 2021)	An Efficient CNN Model for Automated Digital Handwritten Digit Classification	Penelitian ini memfokuskan pada dua batasan masalah CNN yaitu akurasi rendah dan kecepatan komputasi yang lambat. Penelitian mencoba memodelkan arsitektur CNN yang efisien guna memecahkan permasalahan yang ada. Terdapat lima arsitektur CNN yang dimodelkan untuk diterapkan pada dua dataset yang berbeda. Hasil penelitian menunjukkan bahwa dari lima arsitektur yang ada, arsitektur yang dilengkapi <i>stochastic gradient descent</i> dengan <i>momentum optimizer</i> menunjukkan hasil yang terbaik.
<a href="https://e-journal.unair.ac.id/JISEBI/article/download/24237/13992">https://e-journal.unair.ac.id/JISEBI/article/download/24237/13992</a>			Journal of Information Systems Engineering and Business Intelligence
8	(Ali, dkk., 2020)	An Effective and Improved CNN-ELM Classifier for Handwritten Digits Recognition and Classification	Penelitian ini mengintegrasikan antara model CNN dengan algoritma Extreme Learning Machine (ELM). ELM berfungsi untuk mengklasifikasikan gambar dan memiliki kelebihan yaitu kecepatan dalam proses pembelajaran. Hasil penelitian menunjukkan bahwa model tersebut memiliki akurasi lebih baik dibandingkan dengan model CNN konvensional.
<a href="https://www.mdpi.com/2073-8994/12/10/1742/pdf">https://www.mdpi.com/2073-8994/12/10/1742/pdf</a>			<i>Symmetry</i>
9	(Ayumi, dkk., 2016)	Optimization of Convolutional Neural Network using Microcanonical Annealing Algorithm	Penelitian mengoptimasi CNN dengan menggunakan algoritma <i>metaheuristic</i> yaitu algoritma <i>Microcanonical Annealing</i> yang merupakan varian dari <i>Simulated Annealing</i> . Penelitian tersebut mencoba untuk meningkatkan kecepatan komputasi dari proses pembelajaran model. Hasil menunjukkan bahwa model penelitian dapat meningkatkan proses pembelajaran di setiap <i>epoch</i> -nya sampai 4,60%.
<a href="https://ieeexplore.ieee.org/document/7872787">https://ieeexplore.ieee.org/document/7872787</a>			Journal of Theoretical and Applied Information Technology

10	(Zamora, dkk., 2019)	Adaptive Convolutional Neural Network	Penelitian ini menerapkan konsep Adaptive Kernel pada struktur model CNN. Tujuan penerapan Adaptive kernel tersebut untuk mengurangi penggunaan memori pada saat proses pembelajaran model. Pengurangan memori dilakukan dengan cara mengatur ukuran kernel yang digunakan pada <i>layer convolutional</i> dengan ukuran kernel optimal berdasarkan inputan yang digunakan.
<a href="https://openreview.net/forum?id=ByeWdiR5Ym">https://openreview.net/forum?id=ByeWdiR5Ym</a>		ICLR 2019 Conference	

**Tabel II.2 Matriks Penelitian Terkait Penggunaan CNN**

No	Penelit	Algoritma Optimasi							
		SGD	PSO	Diffgrad	SpinalNet	ELM	Adaptive Kernel	Micro. Anealing	Genetic
1	(Prilianti, dkk., 2019)	X							
2	(Syulistyo, dkk., 2016)		X						
3	(Dubey, dkk., 2020)	X							
4	(Lee, dkk., 2021)								X
5	(Andika, dkk., 2019)	X							
6	(Kabir, dkk., 2020)				X				
7	(Biswas & Islam, 2021)	X							
8	(Ali, dkk., 2020)					X			
9	(Ayumi, dkk., 2016)							X	
10	(Zamora, dkk., 2019)						X		

Berdasarkan Tabel II.2 yang merupakan matriks penelitian yang sudah dilakukan. Keterbaharuan penelitian ini yaitu mengoptimisasi penggunaan Adaptive Kernel pada model CNN dengan menggunakan salah satu algoritma Adagrad. Adagrad dipilih karena memiliki kelebihan untuk mengatur ukuran *learning rate* menjadi bagian yang lebih kecil. Semakin kecil *learning rate* yang digunakan, maka diharapkan model penelitian dapat memiliki tingkat akurasi yang semakin tinggi atau meningkat.