

Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)

Journal Homepage: <http://journal.lembagakita.org/index.php/jtik>



Klasifikasi Gambar Pemandangan dengan Kecerdasan Buatan Berbasis CNN

Mozes Hasian Veltin Sinaga ^{1*}, Muhammad Albirra ², Muhammad Fajar Sidiq ³

^{1,2,3} Program Studi Sistem Informasi, Fakultas Informatika, Institut Teknologi Telkom Purwokerto, Kota Purwokerto, Kabupaten Banyumas, Provinsi Jawa Tengah, Indonesia.

article info

Article history:

Received 24 July 2023
Received in revised form
12 January 2024
Accepted 20 March 2024
Available online April 2024.

DOI:

<https://doi.org/10.35870/jti.k.v8i2.1424>.

Keywords:

Artificial Intelligence; CNN;
Classification.

abstract

The use of Artificial Intelligence based on Convolutional Neural Network (CNN) has made remarkable advancements in visual analysis, particularly in landscape image classification. This study applies the CNN method to automatically classify landscape images. Through sophisticated network training and feature extraction steps, CNN can recognize unique patterns and features from various landscape categories, such as mountains, forests, streets, seas, and glaciers. The key advantage of CNN lies in its ability to identify complex and abstract features in images. The evaluation results show that the CNN model achieves satisfying accuracy in classifying landscape images. The application of this method offers practical benefits in various areas, including location recognition, virtual travel, and environmental analysis. AI based on CNN opens new possibilities in visual landscape recognition and its potential to contribute to automated understanding of the beauty of nature.

abstrak

Penggunaan Kecerdasan Buatan berbasis Convolutional Neural Network (CNN) telah mencapai kemajuan pesat dalam analisis visual, khususnya dalam klasifikasi gambar pemandangan. Penelitian ini menerapkan metode CNN untuk mengklasifikasikan secara otomatis gambar-gambar pemandangan alam. Dengan langkah-langkah pelatihan jaringan dan ekstraksi fitur yang canggih, CNN dapat mengenali pola dan fitur unik dari berbagai kategori pemandangan, seperti gunung, hutan, jalan, laut, dan gletser. Keunggulan utama CNN terletak pada kemampuannya untuk mengidentifikasi fitur-fitur kompleks dan abstrak dalam gambar. Hasil evaluasi menunjukkan bahwa model CNN mencapai akurasi yang memuaskan dalam mengklasifikasikan gambar pemandangan. Penggunaan metode ini menawarkan manfaat praktis dalam berbagai aplikasi, termasuk pengenalan lokasi, perjalanan virtual, dan analisis lingkungan. Kecerdasan Buatan berbasis CNN membuka peluang baru dalam pengenalan visual alam dan potensinya untuk berkontribusi dalam memahami keindahan alam secara otomatis.

Corresponding Author. Email: mozzessinaga007@gmail.com ^{1}.

© E-ISSN: 2580-1643.

Copyright @ 2024 by the authors of this article. Published by Lembaga Otonom Lembaga Informasi dan Riset Indonesia (KITA INFO dan RISET). This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

1. Latar Belakang

Kecerdasan Buatan (*Artificial Intelligence/AI*) telah mencapai kemajuan yang pesat dalam beberapa tahun terakhir [1]. Salah satu aspek penting dalam pengembangan AI adalah kemampuannya untuk memahami dan mengklasifikasikan gambar. Salah satu metode yang paling sukses dalam klasifikasi gambar adalah menggunakan Jaringan Saraf Tiruan Konvolusi (*Convolutional Neural Network/CNN*) [2]. Pemandangan alam memiliki keindahan yang luar biasa dan menjadi subjek yang populer dalam fotografi dan seni visual. Mengklasifikasikan gambar pemandangan alam secara otomatis dapat membantu dalam berbagai aplikasi, termasuk pengenalan lokasi, perjalanan virtual, identifikasi objek dalam gambar, dan banyak lagi [3]. Jaringan Saraf Tiruan Konvolusi (CNN) adalah salah satu jenis arsitektur yang paling sukses dalam pengenalan gambar. CNN dirancang secara khusus untuk memahami pola dan fitur dalam gambar. Arsitektur ini terinspirasi oleh cara kerja visual manusia, di mana kita mengidentifikasi objek berdasarkan pola dan fitur spesifik [4].

Penggunaan CNN dalam klasifikasi gambar pemandangan alam melibatkan beberapa tahap. Pertama, CNN dilatih menggunakan sejumlah besar data gambar pemandangan yang telah dikategorikan sebelumnya. Selama pelatihan, jaringan belajar mengenali pola dan fitur khas dari setiap kategori pemandangan, seperti pegunungan, pantai, hutan, dan lainnya [5]. Setelah melalui tahap pelatihan, CNN dapat digunakan untuk mengklasifikasikan gambar pemandangan yang tidak diketahui. Proses ini melibatkan ekstraksi fitur dari gambar menggunakan lapisan konvolusi, di mana jaringan menemukan pola dan fitur-fitur penting. Kemudian, fitur-fitur ini diumpulkan ke lapisan-lapisan berikutnya, yang secara bertahap mengklasifikasikan gambar ke dalam kategori yang sesuai berdasarkan apa yang telah dipelajari selama pelatihan [6].

Keunggulan CNN dalam klasifikasi gambar pemandangan adalah kemampuannya untuk mengenali fitur-fitur yang sangat kompleks dan abstrak. Misalnya, CNN dapat membedakan antara berbagai jenis awan, mengidentifikasi struktur dan tekstur yang berbeda pada pegunungan, serta mengenali ciri-ciri khas pantai seperti pasir, ombak, dan langit biru. Penggunaan CNN dalam klasifikasi

gambar pemandangan tidak hanya memberikan manfaat praktis dalam pengenalan objek dan kategori pemandangan, tetapi juga menghadirkan potensi baru dalam berbagai bidang seperti pariwisata, pemetaan, dan analisis lingkungan. Dalam pengembangan masa depan, teknik-teknik AI berbasis CNN akan terus ditingkatkan untuk meningkatkan akurasi dan keandalan dalam klasifikasi gambar pemandangan [7]. Dengan demikian, dalam penelitian ini, kita akan mengeksplorasi penggunaan Kecerdasan Buatan berbasis CNN untuk klasifikasi gambar pemandangan alam. Melalui pelatihan jaringan dan ekstraksi fitur yang canggih, kita akan mencari cara untuk meningkatkan kemampuan AI dalam mengenali dan mengklasifikasikan pemandangan alam secara otomatis.

2. Metode Penelitian

Klasifikasi adalah pendekatan analisis yang digunakan untuk memprediksi atau mengklasifikasikan data ke dalam beberapa kategori atau kelas berdasarkan atribut atau fitur yang dimiliki. Metode ini bertujuan untuk mengidentifikasi pola atau relasi di antara data yang dapat digunakan untuk memperoleh pemahaman yang lebih baik tentang hubungan antar variable [8]. Salah satu metode klasifikasi yang umum digunakan adalah *Convolutional Neural Network (CNN)*. CNN awalnya dikembangkan untuk analisis dan pemrosesan data visual, terutama dalam pengenalan gambar. Namun, metode CNN juga telah diperluas untuk digunakan dalam berbagai bidang seperti pengolahan teks, pengenalan suara, dan analisis data lainnya [9].

Metode Convolutional Neural Network (CNN)

Metode *Convolutional Neural Network (CNN)* adalah salah satu teknik dalam bidang kecerdasan buatan yang digunakan untuk analisis dan pemrosesan data visual, seperti gambar dan video. CNN memiliki kemampuan untuk secara otomatis mengekstraksi fitur-fitur yang relevan dari data visual tersebut dan menggunakan untuk melakukan klasifikasi [10]. Langkah-langkah dalam menggunakan metode CNN sebagai berikut:

1) Persiapan Data

Mengumpulkan dan menyusun data yang akan digunakan untuk pelatihan dan pengujian model

CNN. Data tersebut harus memiliki label atau kategori yang jelas untuk proses klasifikasi.

2) Pembentukan Arsitektur CNN

Menentukan arsitektur atau struktur jaringan CNN yang akan digunakan. Arsitektur CNN terdiri dari beberapa layer, seperti layer konvolusi, layer pooling, dan layer *fully connected*.

3) Pelatihan Model

Melakukan pelatihan model CNN menggunakan data pelatihan yang telah disiapkan. Proses pelatihan melibatkan pengoptimalan parameter-parameter model untuk meminimalkan kesalahan prediksi.

4) Validasi dan Evaluasi

Menggunakan data validasi untuk menguji performa model CNN yang telah dilatih. Mengukur metrik evaluasi, seperti akurasi, presisi, recall, dan F1-score, untuk mengevaluasi performa model.

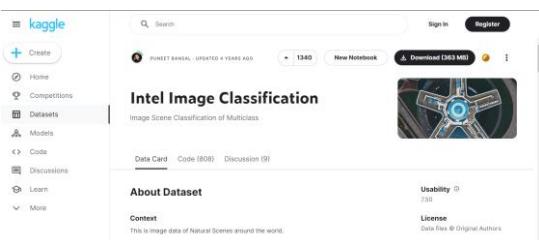
5) Prediksi

Setelah model CNN dilatih dan dievaluasi, model tersebut dapat digunakan untuk melakukan prediksi atau klasifikasi pada data baru yang belum pernah dilihat sebelumnya.

Dengan menggunakan metode *Convolutional Neural Network* (CNN), penelitian ini dapat mengklasifikasikan data visual dengan akurasi yang tinggi, seperti gambar produk atau atribut visual lainnya, untuk tujuan analisis dan pemahaman yang lebih mendalam.

3. Hasil dan Pembahasan

Persiapan data dilakukan dengan pengambilan data menggunakan Kaggle. Kaggle adalah sebuah platform kompetisi dan sumber daya berbasis online yang memfasilitasi dan menyediakan akses untuk belajar, berlatih, dan berkompetisi dalam berbagai masalah analisis data dan ilmu data.



Gambar 1. Dataset Kaggle

Pembentukan Arsitektur CNN

Convolutional Neural Network (CNN) merupakan algoritma deep learning yang digunakan untuk mengolah input data berupa gambar. Algoritma ini bertujuan untuk menentukan bobot dan bias yang dapat dipelajari untuk mengidentifikasi berbagai aspek dalam gambar, dan secara khusus berfungsi untuk membedakan objek satu dengan objek lainnya. CNN adalah salah satu jenis arsitektur *neural network* yang umum digunakan untuk tugas-tugas analisis data berbasis gambar. Meskipun demikian, CNN ini pada dasarnya tidak berbeda secara signifikan dengan neural network pada umumnya.

```
#CNN Network

class ConvNet(nn.Module):
    def __init__(self,num_classes=6):
        super(ConvNet,self).__init__()

        #Output size after convolution filter
        #((w-f+2P)/s)+1

        #Input shape= (256,3,150,150)
        self.conv1=nn.Conv2d(in_channels=3,out_channels=12,kernel_size=3,stride=1,padding=1)
        #Shape= (256,12,150,150)
        self.bn1=nn.BatchNorm2d(num_features=12)
        #Shape= (256,12,150,150)
        self.relu1=nn.ReLU()
        #Shape= (256,12,150,150)

        self.pool1=nn.MaxPool2d(kernel_size=2)
        #Reduce the image size by factor 2
        #Shape= (256,12,75,75)

        self.conv2=nn.Conv2d(in_channels=12,out_channels=20,kernel_size=3,stride=1,padding=1)
        #Shape= (256,20,75,75)
        self.relu2=nn.ReLU()
        #Shape= (256,20,75,75)

        self.conv3=nn.Conv2d(in_channels=20,out_channels=32,kernel_size=3,stride=1,padding=1)
        #Shape= (256,32,75,75)
        self.bn3=nn.BatchNorm2d(num_features=32)
        #Shape= (256,32,75,75)
        self.relu3=nn.ReLU()
        #Shape= (256,32,75,75)

        self.fc=nn.Linear(in_features=75 * 75 * 32,out_features=num_classes)
```

Gambar 2. Arsitektur CNN bagian 1

```
#Feed forward function

def forward(self,input):
    output=self.conv1(input)
    output=self.bn1(output)
    output=self.relu1(output)

    output=self.pool1(output)

    output=self.conv2(output)
    output=self.relu2(output)

    output=self.conv3(output)
    output=self.bn3(output)
    output=self.relu3(output)

    #Above output will be in matrix form, with shape (256,32,75,75)
    output=output.view(-1,32*75*75)

    output=self.fc(output)

    return output
```

Gambar 3. Arsitektur CNN bagian 2

Pelatihan Model

Pelatihan model CNN AI adalah proses mengajarkan jaringan saraf tiruan untuk mengenali pola dalam data gambar. Dilakukan dengan dataset gambar dan label, model disesuaikan dan diperbarui berulang kali menggunakan algoritma optimisasi seperti *Stochastic Gradient Descent* (SGD). Tujuannya adalah mencapai kinerja yang baik dalam klasifikasi, deteksi, atau tugas lain berbasis gambar.

```
import torch
from torch.autograd import Variable
import matplotlib.pyplot as plt

best_accuracy = 0.0
train_accuracy_list = []
test_accuracy_list = []
train_loss_list = []

for epoch in range(num_epochs):
    # Evaluation and training on training dataset
    model.train()
    train_accuracy = 0.0
    train_loss = 0.0

    for i, (images, labels) in enumerate(train_loader):
        if torch.cuda.is_available():
            images = Variable(images.cuda())
            labels = Variable(labels.cuda())

        optimizer.zero_grad()

        outputs = model(images)
        loss = loss_function(outputs, labels)
        loss.backward()
        optimizer.step()

        train_loss += loss.cpu().data * images.size(0)
        _, prediction = torch.max(outputs.data, 1)
        train_accuracy += int(torch.sum(prediction == labels.data))

    train_accuracy = train_accuracy / train_count
    train_loss = train_loss / train_count

    # Evaluation on testing dataset
    model.eval()

    test_accuracy = 0.0
```

Gambar 4. Code untuk Pelatihan Data bagian 1

```
test_accuracy = 0.0
for i, (images, labels) in enumerate(test_loader):
    if torch.cuda.is_available():
        images = Variable(images.cuda())
        labels = Variable(labels.cuda())

    outputs = model(images)
    _, prediction = torch.max(outputs.data, 1)
    test_accuracy += int(torch.sum(prediction == labels.data))

test_accuracy = test_accuracy / test_count

print('Epoch: ' + str(epoch) + ' Train Loss: ' + str(train_loss) + ' Train Accuracy: ' + str(train_accuracy) + ' Test Accuracy: ' + str(test_accuracy))

# Save the best model
if test_accuracy > best_accuracy:
    torch.save(model.state_dict(), '/content/drive/My Drive/BSI1/test_checkpoint.model')
    best_accuracy = test_accuracy

# Append accuracy and loss to the respective lists
train_accuracy_list.append(train_accuracy)
test_accuracy_list.append(test_accuracy)
train_loss_list.append(train_loss)

Epoch: 0 Train loss: tensor(0.4483) Train Accuracy: 0.6913253527348354 Test Accuracy: 0.5996666666666667
Epoch: 1 Train loss: tensor(0.8239) Train Accuracy: 0.721686074362265 Test Accuracy: 0.69
Epoch: 2 Train loss: tensor(0.2718) Train Accuracy: 0.8033333333333334 Test Accuracy: 0.7333333333333334
Epoch: 3 Train loss: tensor(0.4848) Train Accuracy: 0.8501998670293145 Test Accuracy: 0.743
Epoch: 4 Train loss: tensor(0.1771) Train Accuracy: 0.8961818819069806 Test Accuracy: 0.753
Epoch: 5 Train loss: tensor(0.2718) Train Accuracy: 0.9160677698002 Test Accuracy: 0.7336666666666667
Epoch: 6 Train loss: tensor(0.1771) Train Accuracy: 0.9451332478378706 Test Accuracy: 0.769
Epoch: 7 Train loss: tensor(0.1626) Train Accuracy: 0.9511889072245057 Test Accuracy: 0.7666666666666667
Epoch: 8 Train loss: tensor(0.1626) Train Accuracy: 0.9511889072245057 Test Accuracy: 0.7673333333333333
Epoch: 9 Train loss: tensor(0.1626) Train Accuracy: 0.9511889072245057 Test Accuracy: 0.7666666666666667
```

Gambar 5 Code untuk pelatihan data bagian 2

Validasi dan Evaluasi

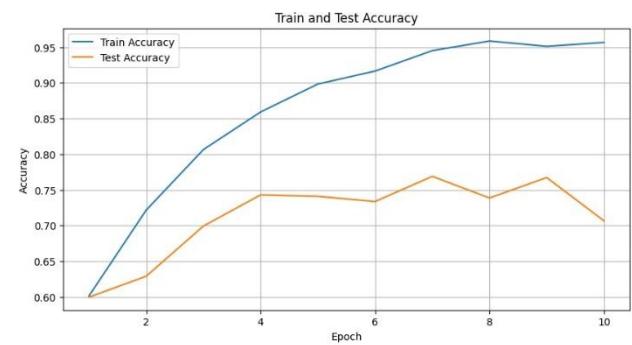
Akurasi merupakan metrik yang digunakan untuk mengevaluasi performa dari model klasifikasi. Metrik ini mencerminkan persentase prediksi yang tepat dari model dibandingkan dengan total prediksi yang dilakukan.

```
import matplotlib.pyplot as plt

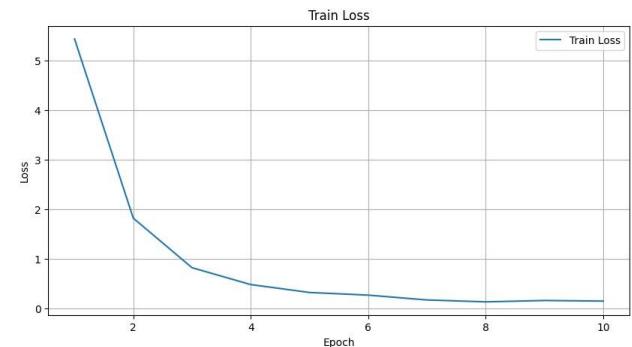
# Plot accuracy
plt.figure(figsize=(10, 5))
plt.plot(range(1, num_epochs + 1), train_accuracy_list, label='Train Accuracy')
plt.plot(range(1, num_epochs + 1), test_accuracy_list, label='Test Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.title('Train and Test Accuracy')
plt.legend()
plt.grid(True)
plt.show()

# Plot loss
plt.figure(figsize=(10, 5))
plt.plot(range(1, num_epochs + 1), train_loss_list, label='Train Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.title('Train Loss')
plt.legend()
plt.grid(True)
plt.show()
```

Gambar 6. Code untuk mengukur tingkat akurasi



Gambar 7. Grafik Train and Test Accuracy



Gambar 8. Grafik Train Loss

Hasil Prediksi

Hasil klasifikasi menggunakan *Convolutional Neural Network* (CNN) untuk klasifikasi pemandangan dengan kategori mountain, forest, street, sea, dan glacier adalah prediksi kategori untuk setiap gambar pemandangan yang diberikan. Model CNN telah dilatih dengan menggunakan dataset yang berisi berbagai gambar pemandangan yang sudah diberi label sesuai dengan kategori yang ditentukan.

Setelah melalui proses pelatihan yang dilakukan oleh CNN, model tersebut telah belajar mengekstraksi fitur-fitur visual yang relevan dari gambar

pemandangan. Kemudian, model tersebut dapat memproses gambar pemandangan baru dan menghasilkan prediksi kategori berdasarkan fitur-fitur yang telah dipelajari. Hasil klasifikasi ditampilkan dalam bentuk kategori seperti mountain, forest, street, sea, atau glacier, yang menunjukkan kategori pemandangan yang paling sesuai dengan gambar yang diberikan. Akurasi dan performa model dapat diukur dengan menghitung persentase prediksi yang benar terhadap keseluruhan gambar pemandangan yang diuji.

```

❶ #prediction function
❷ def prediction(img_path,transformer):
    image=Image.open(img_path)
    image_tensor=transformer(image).float()

    image_tensor=image_tensor.unqueeze_(0)
    if torch.cuda.is_available():
        image_tensor.cuda()
    input=Variable(image_tensor)

    output=model(input)
    index=output.data.numpy().argmax()
    pred=classes[index]

    return pred

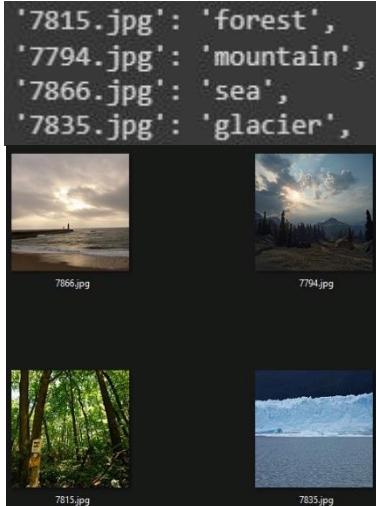
❸ images_path=glob(pred_path+'*.jpg')

❹ pred_dict={}
    for i in images_path:
        pred_dict[i[i.rfind('/')+1:]] = prediction(i,transformer)

❺ pred_dict

```

Gambar 9. Code untuk memprediksi data gambar



Gambar 10. Hasil dari prediksi

4. Kesimpulan

Setelah melakukan pelatihan model menggunakan dataset pemandangan yang telah disiapkan, kami melakukan validasi dan evaluasi untuk mengukur performa model. Hasil evaluasi menunjukkan bahwa model CNN mampu mengklasifikasikan gambar pemandangan dengan akurasi yang baik.

Penerapan metode CNN dalam klasifikasi gambar pemandangan alam memiliki potensi untuk memberikan manfaat dalam berbagai aplikasi, seperti pengenalan lokasi, perjalanan virtual, analisis lingkungan, dan banyak lagi. Keberhasilan metode ini dalam mengklasifikasikan gambar pemandangan menunjukkan kemajuan pesat dalam pengembangan kecerdasan buatan dan pengenalan gambar. Potensi besar penggunaan Kecerdasan Buatan berbasis CNN dalam klasifikasi gambar pemandangan alam dan memberikan pandangan bagaimana AI dapat berkontribusi dalam pemahaman dan pengenalan visual dunia alam.

5. Ucapan Terima Kasih

Terima kasih atas kontribusi penelitian yang telah dilakukan dalam menggunakan metode *Convolutional Neural Network* (CNN) untuk mengklasifikasikan gambar pemandangan alam dengan akurasi yang baik. Penelitian ini telah membuktikan potensi besar Kecerdasan Buatan dalam mengenali dan mengklasifikasikan pemandangan alam secara otomatis. Kami mengapresiasi upaya dalam mengembangkan arsitektur CNN dan melatih model dengan dataset yang relevan. Hasil prediksi yang diperoleh menunjukkan kemajuan pesat dalam pengenalan visual, dan kami percaya bahwa metode ini memiliki manfaat praktis yang luas, seperti dalam aplikasi pariwisata, pemetaan, dan analisis lingkungan. Terima kasih atas kontribusi berharga untuk para dosen dan rekan-rekan kami dalam memajukan bidang kecerdasan buatan dan analisis gambar. Semoga penelitian ini menjadi landasan untuk pengembangan lebih lanjut dalam memanfaatkan teknologi AI dalam memahami dan mengapresiasi keindahan alam. Sukses selalu dalam penelitian dan pengembangan teknologi masa depan.

6. Daftar Pustaka

- [1] Li, J., Zhang, H., Li, D., & Chen, H. (2015). On the performance of wireless-energy-transfer-enabled massive MIMO systems with superimposed pilot-aided channel estimation. *IEEE Access*, 3, 2014-2027. DOI: 10.1109/ACCESS.2015.2492780.

- [2] Qin, J., Pan, W., Xiang, X., Tan, Y., & Hou, G. (2020). A biological image classification method based on improved CNN. *Ecological Informatics*, 58, 101093. DOI: <https://doi.org/10.1016/j.ecoinf.2020.10109>.
- [3] Azmi, K., Defit, S., & Sumijan, S. (2023). Implementasi convolutional neural network (CNN) untuk klasifikasi batik tanah liat sumatera barat. *Jurnal Unitek*, 16(1), 28-40.
- [4] Xin, R., Zhang, J., & Shao, Y. (2020). Complex network classification with convolutional neural network. *Tsinghua Science and technology*, 25(4), 447-457.
- [5] Zhang, P., Wang, X., Zhang, W., & Chen, J. (2018). Learning spatial-spectral-temporal EEG features with recurrent 3D convolutional neural networks for cross-task mental workload assessment. *IEEE Transactions on neural systems and rehabilitation engineering*, 27(1), 31-42. DOI: 10.1109/TNSRE.2018.2884641.
- [6] Lu, Y., Lu, G., Lin, R., Li, J., & Zhang, D. (2019). SRGC-Nets: Sparse repeated group convolutional neural networks. *IEEE transactions on neural networks and learning systems*, 31(8), 2889-2902. DOI: 10.1109/TNNLS.2019.2933665.
- [7] Lin, B. S., Cheng, J. S., Liao, H. C., Yang, L. W., Yang, T., & Chen, K. C. (2021). Improvement of multi-lines bridge defect classification by hierarchical architecture in artificial intelligence automatic defect classification. *IEEE Transactions on Semiconductor Manufacturing*, 34(3), 346-351. DOI: 10.1109/TSM.2021.3076808.
- [8] Qaffas, A. A., Hajkacem, M. B., Ncir, C. B., & Nasraoui, O. (2023). Interpretable Multi-criteria ABC analysis based on semi-supervised clustering and Explainable Artificial Intelligence. *IEEE Access*. DOI: 10.1109/ACCESS.2023.3272403.
- [9] Yuan, L., Wei, X., Shen, H., Zeng, L. L., & Hu, D. (2018). Multi-center brain imaging classification using a novel 3D CNN approach. *IEEE Access*, 6, 49925-49934. DOI: 10.1109/ACCESS.2018.2868813.