Journal of New Results in Engineering and Natural Science, No:21 (2024) 1-17
https://dergipark.org.tr/tr/pub/jrens

Received: 06.12.2024
*Accepted:*18.12.2024

Identification and Classification of Damage in DNA Imagery Using Deep Learning Algorithms

Cengiz GÜNGÖR[1*], Ali AKTAŞ[2]

[1]Gaziosmanpasa University, Faculty of Engineering and Natural Sciences, Department of Computer Engineering, 60250, Tokat, Turkey
[2]Gaziosmanpasa University, Faculty of Engineering and Natural Sciences, Mechatronics Engineering Department, 60250, Tokat, Turkey

Abstract

In this study, the application of deep learning, particularly Convolutional Neural Networks (CNNs), to analyze comet assay images for DNA damage assessment is explored. The comet as-say is a pivotal method for detecting DNA strand breaks at the cellular level, essential in geno-toxicity and carcinogenicity research. Traditional approaches to analyze these images often in-volve manual labor or basic computational tools, which are inefficient, especially with noisy data. This research addresses these inefficiencies by developing a custom CNN model to auto-matically classify DNA damage levels in comet assay images. The dataset consists of 5,326 im-ages, categorized into six damage levels: from undamaged (C0) to extensively damaged (C4), plus an unidentifiable category (C6). Data augmentation was employed to enhance the model's robustness by creating varied inputs for training. The CNN processes the raw images through several layers to extract features and identify patterns, facilitating the classification of DNA damage levels. The model's performance was assessed using a confusion matrix, achieving an overall classification accuracy of approximately 92%. Although the model was highly accurate in distinguishing severe damage levels, it struggled with closely related classes, such as slightly and moderately damaged DNA. This study underscores the potential of deep learning in auto-mating and improving the analysis of comet assay images. CNNs offer a more accurate and effi-cient alternative to traditional methods, which could significantly advance research in genotoxi-city and clinical diagnostics, leading to a better understanding and monitoring of DNA damage in biological systems.

Keywords: The single-cell gel electrophoresis; Deep learning; Comet assay

Derin Öğrenme Algoritmaları ile DNA Görüntülerindeki Hasarın Tanımlanması ve Sınıflandırılması

Cengiz GÜNGÖR[1*], Ali AKTAŞ[2]

Özet

Komet testi görüntülerindeki DNA hasarını değerlendirmek için derin öğrenme, özellikle de Evrişimli Sinir Ağları (CNN) uygulaması incelenmiştir. Komet testi, hücresel düzeyde DNA kırılmalarını tespit etmek için kullanılan ve genotoksikite ve kanserojenite araştırmalarında temel bir yöntemdir. Bu görüntülerin geleneksel analiz yöntemleri genellikle manuel işlemler veya basit hesaplamalı araçlar içermekte olup, özellikle gürültülü verilerde verimsizdir. Bu çalışma, komet testi görüntülerindeki DNA hasar seviyelerini otomatik olarak sınıflandırmak için özel bir CNN modeli geliştirerek bu

[*] *Corresponding Author*, e- mail: cengiz.gungor@gop.edu.tr

verimsizliklerin üstesinden gelmeyi amaçlamıştır. Veri seti, hasarsız (C0) durumdan yoğun hasarlı (C4) duruma kadar altı farklı hasar seviyesi ve tanımlanamayan bir kategori (C6) olmak üzere toplam 5.326 görüntüden oluşmaktadır. Modelin sağlamlığını artırmak için veri genişletme tekniği kullanılmıştır. CNN, ham görüntüleri çeşitli katmanlardan geçirerek özellik çıkarma ve desen tanıma yaparak DNA hasar seviyelerinin sınıflandırılmasını kolaylaştırır. Modelin performansı, bir karışıklık matrisi kullanılarak değerlendirilmiş ve yaklaşık %92 genel sınıflandırma doğruluğuna ulaşmıştır. Model, şiddetli hasar seviyelerini ayırt etmede oldukça başarılı olsa da, hafif ve orta derecede hasarlı DNA gibi yakın sınıflarda zorluk yaşamıştır. Bu çalışma, derin öğrenmenin komet testi görüntülerinin analizini otomatize etme ve iyileştirme potansiyelini vurgulamaktadır. CNN'ler, genotoksikite ve klinik teşhis alanındaki araştırmalara önemli katkılar sağlayabilecek daha doğru ve verimli bir alternatif sunarak biyolojik sistemlerdeki DNA hasarının daha iyi anlaşılmasına ve izlenmesine olanak tanıyabilir.

Anahtar Kelimeler : Tek Hücreli Jel Elektroforezi, Derin Öğrenme, Komet Testi

1. Introduction

The single-cell gel electrophoresis test, commonly known in the literature as the "comet assay," is one of the most frequently used methods in molecular and cellular biology for detecting DNA damage. Developed by Ostling and Johansson in 1984, this test has been refined over the years to accurately measure DNA fragmentation levels. DNA plays a critical role in drug development and toxicology, and concerns about DNA damage are significant due to its potential to lead to various diseases, including cancer and chronic conditions. The comet assay or single-cell gel electrophoresis is recommended for detecting DNA strand breaks. In this assay, damaged DNA migrates out of the nucleus, forming a comet-like tail while undamaged DNA remains in a circular shape. The comet assay is highly sensitive, efficient, and cost-effective, providing researchers with extensive information. It is utilized to evaluate the genotoxicity and carcinogenic potential of compounds in toxicology, understand disease pathogenesis in medical research, and monitor disease conditions in clinical medicine [1-7].

DNA damage in comet assay images can be determined through visual scoring or computer image analysis tools. Various image analysis programs, such as CometQ, OpenComet, and HiComet, have been developed to automate the analysis process. These tools use predefined methods to segment and assess comet scores; however, they all rely on manually labeled image features and general machine learning techniques, such as support vector machines. Manual labeling and feature adjustment are labor-intensive and time-consuming tasks that limit the efficiency of these methods, especially when dealing with noisy images and multiple aspect ratios. These tools enhance the efficiency and impartiality of evaluating DNA damage, providing valuable insights for research in toxicology, medical research, and clinical medicine. Neural networks have the capability to process raw images and can be particularly successful in detecting and scoring comet tails through an end-to-end learning process. These networks can often be trained using transfer learning on large, general image datasets, enabling them to surpass manually engineered features [6-11].

An evaluation of the literature on DNA studies reveals that the number of studies utilizing deep learning methods is quite limited. Since 1992, only 17,000 publications have been indexed in the Web of Science database, with a similar number found in the Scopus database. Typically, these publications focus on assessing the results of the comet assay method to examine different cell types and DNA damage. The existing studies emphasize the continuous classification of images by expert researchers and the evaluation of the results. An analysis of the Web of Science and Scopus databases concerning comet assay and deep learning studies indicates that there are very few studies available. The distribution of these studies by country, as shown in Figure 1, illustrates the geographic distribution of comet assay-related research indexed in the Web of Science database [12-16].

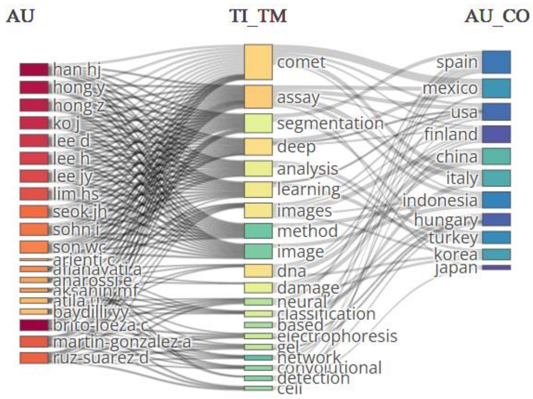


Figure 1. *Studies on the Comet Assay in conjunction with deep learning in the Web of Science database.*

When the literature studies are examined in detail, we evaluated the following studies as important. Sreelatha et al. developed CometQ, an automated tool for detecting and measuring DNA damage using comet assay images. They primarily used the Support Vector Machine (SVM) algorithm to distinguish between silver-stained and fluorescent-stained images, and to classify the comet types into weak, moderate, or severe damage. This study utilized over 600 images, achieving a positive predictive value of 90.26% and a sensitivity of 93.34% [17]. Lee et al. developed a method to classify DNA damage patterns in comet assay images. RGB images were converted to grayscale, followed by preprocessing steps including scale bar correction, mean filtering, and moving average filtering. The Canny edge detector algorithm was used to eliminate overlapping comets. For 20 test datasets containing over 300 images, the average classification accuracy was calculated to be 86.80% [18]. Gyori et al. introduced OpenComet, an open-source software for the automatic analysis of gel electrophoresis images used in comet imaging experiments. The image processing involved adaptive thresholding based on intensity histograms to obtain binary images, where black pixels represent the background and white pixels denote comets. The brightest regions were assumed to be the comet heads, and intensity profile analysis was used to separate the head and tail regions. The system's sensitivity was calculated at 63.95% [11]. The MelNet model, developed using the KITTI dataset, was tested alongside EfficientDet, YoloV5, and MobileNetV3. Upon evaluation, the MelNet architecture achieved the highest class accuracy value of 95.52%, depending on different epoch values [53]. Sreelatha et al. also proposed a method for effectively and fully automatically detecting comets in noisy silver-stained comet assay images. Their software measured parameters such as comet length, tail length, head diameter, percentage of DNA in the head, percentage of DNA in the tail, and tail moment. The analysis consisted of three stages: comet detection, comet segmentation, and comet measurement. Shading correction was applied to correct images, and contrast enhancement was used to distinguish comets from the background. Gaussian filtering was applied to smooth images and obtain elongated spots. Otsu thresholding was then used to obtain binary images. The detected objects' contours were refined using morphological closing. The system's sensitivity was calculated at 89.30% [19]. Sansone et al. proposed an algorithm for automatic comet analysis. The study was conducted in two stages: comet detection and comet segmentation. Gaussian pre-filtering and morphological operators were used for comet detection in the first stage, and fuzzy clustering was used for comet segmentation in the second stage. High sensitivity results were obtained in both stages, with an overall sensitivity of 78.96% [20]. Böcker et al. developed a system based on specially developed software and hardware for the automatic analysis of DNA damage, which also requires human interaction. The analysis was divided into two parts: automatic cell recognition and comet classification, and comet measurement. Mathematical morphology-based algorithms were used in the preprocessing, segmentation, and feature classification stages. The histogram analysis, entropy maximization, k-means clustering, and contour-based procedures reported in the study failed. Therefore, two procedures were used to calculate the threshold value and analyze the image content. Adaptive thresholding was used to improve segmentation results. Parallel programming was utilized in their software to achieve faster results. The system's sensitivity was calculated at 95.20% and specificity at 92.70% [21]. Gonzalez et al. developed a software program called CellProfiler for automatically identifying, measuring, and extracting comet assay information. The Gaussian mixture model and expectation-maximization algorithm were used to identify comets. The methods used in the study were not explicitly explained. The performance of the CellProfiler software was mainly compared with the CASP software [22]. Rosati et al. conducted a review comparing deep learning models trained on publicly available datasets for comet damage detection. They achieved a sensitivity of 74% from the training process [23]. Attila et al. developed an application using existing classifier methods in MATLAB for special analysis of comet images. The success rate achieved was around 68% [24]. Srikanth et al. used the VGG19 architecture from the transfer learning structure in their deep learning model. By changing the hyperparameter values on the model, different test results ranging from 39% to 84% were obtained [25]. Afiahayati et al. used the Faster R-CNN object detection algorithm in their study and achieved a 95% success rate using the ResNet50 and ResNet101 architectures, which were trained on the COCO dataset using the transfer learning method [26].

1.1. Deep learning

Deep learning is a machine learning technique that enables computer systems to maximize the use of experience hidden in data, work with complex real-world data, handle these complex data with nested hierarchies, and achieve more successful results by defining these hierarchies with simple relationships. According to another definition, deep learning is a machine learning technique that combines artificial neural networks, artificial intelligence, graphical modeling, optimization, model recognition, and signal processing to make more powerful predictions [27]. In deep learning, multi-layered machine learning models apply non-linear transformations to the data at each layer and transfer it to a higher, more abstract layer, where supervised or unsupervised learning takes place. The performance of a machine learning technique greatly depends on the good representation of input data. Therefore, preprocessing of data is a critical step in the process of creating learning systems. During the feature extraction process, experts work to reduce the dimensions of the input data's features. The performance of shallow learning models like Support Vector Machines (SVMs) and logistic regression depends on this feature extraction process. This process is important but also very time-consuming and exhausting. Solving this problem with algorithms that simplify the task is a more practical approach. Deep learning techniques are one of the best solutions for handling high-dimensional data and extracting distinguishing information from data. Deep learning algorithms have the ability to automate feature extraction without needing expert knowledge [28]. Since deep learning is a type of machine learning, the basic principles of traditional machine learning have influenced the development of deep networks [29]. Components such as the activation function and optimization algorithm play a crucial role in producing accurate results in the efficient and effective training of a deep learning model [30].

1.2. Activation Functions

In biological neurons, signals coming from dendrites accumulate in the cell body, and if the resulting signal strength exceeds a certain threshold, it is transmitted to the output, i.e., the axon; otherwise, it is not transmitted. Similarly, in an artificial neuron, this task is performed by the activation function. The activation function decides whether to transmit the signal [31]. Commonly used activation functions in the literature are given in Figure 2.

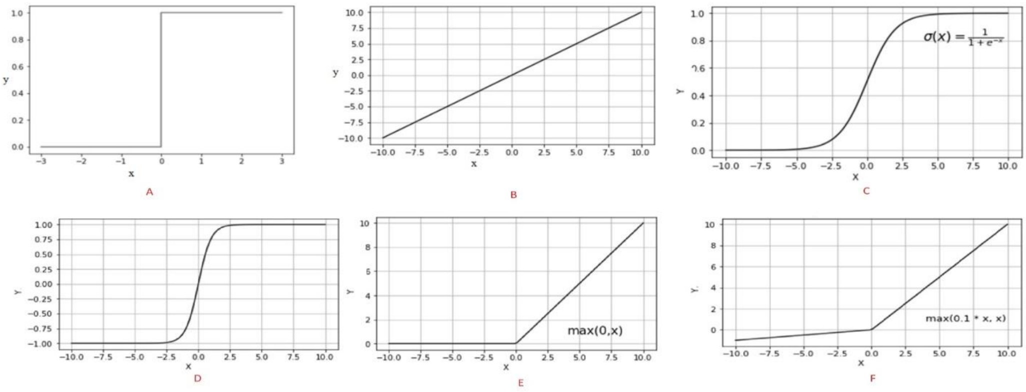


Figure 2. *Activation Functions*

The step activation function, a simple step function commonly used for a single neuron model, produces values of “1” or “0” based on the threshold. If the input is less than zero, it produces “0”; otherwise, it produces “1”. As seen in Figure 2-A, the function loses its distinctiveness because the input takes the value of 0. Moreover, it is not possible to use this function to update weights since it only produces values of 0 or 1. Therefore, it is not used in deep learning algorithms [32]. The linear activation function is defined as $\mathrm{f(x)} = \mathrm{Wx}$ or $\hat{\mathrm{y}} = W^T\mathrm{x} + \mathrm{b}$. Here, $\hat{\mathrm{y}}$ is the output produced by the network, W^T is the weight vector, $\mathbf{x}$ is the independent variables or network inputs, and bbb is the bias. As seen in Figure 2-B, the dependent variable changes proportionally with the independent variable. The linear activation function is effectively used in shallow architectures, but its use in deep learning problems is limited [33]. The sigmoid activation function, also known as the logistic activation function, converts the independent variables of the infinite range into simple probabilities between 0 and 1 [33]. The graph of the sigmoid activation function is shown in Figure 2-C [29]. The hyperbolic tangent activation function, widely used for artificial neural networks that need to produce values between -1 and 1, is shown in Figure 2-D. Its shape is similar to the sigmoid function, but it has the advantage of allowing negative numbers. However, like the sigmoid function, the outputs produced are not zero-centered and it faces the problem of vanishing gradients [34]. The softmax activation function is a generalization of logistic regression that can be applied to continuous data instead of binary classification. Since it produces output for more than two classes, it is usually located in the output layer of a classifier. If a high number of classes, such as a thousand, are predicted for a classifier, a variant of the softmax activation function, hierarchical softmax, is used. The hierarchical softmax activation function separates the thousands of predicted classes using a tree structure [33]. Recently, a simple function called the Rectified Linear Unit (ReLU) has become very popular as it produces very good experimental results. The ReLU activation function is defined as $\boldsymbol{f}(x) = \max{(0, x)}$ As seen in Figure 2-E, it is zero for negative values and increases linearly for positive values [35]. The Leaky ReLU activation function is used to address the dying ReLU problem. The Leaky ReLU activation function provides a small gradient when the neuron is not active. The Leaky ReLU activation function is defined as shown in Figure 2-F [36].

1.3. Optimization Algorithms

Machine learning relies on techniques that minimize errors in equations through optimization. In optimization, the focus is on conditions and parameters that produce predictions closest to expected values. Adjusting weights to produce better predictions is known as parameter optimization. Similar to the scientific method development process, hypotheses are repeatedly altered and tested until the best description for real-world events is found. Each set of weights represents a hypothesis indicating the significance of inputs. Weights are assumptions about the correlation between the inputs and the desired outputs of the network. Optimization plays a central role in applied mathematics and is widely used for modeling real-world problems. It encompasses complex systems in fields such as image science, finance, signal processing, and machine learning. Most optimization algorithms are iterative, meaning they perform successive calculations to converge on the desired solution. In programming, loops are used for iteration. In the $\boldsymbol{i}$'th iteration, the value of the input $\boldsymbol{x_i}$ is calculated. The loop terminates only when a convergence criterion or stopping criterion is met [37]. In deep learning, commonly used optimization algorithms are gradient-based algorithms, including: Gradient Descent, Gradient Descent with Momentum, Nesterov Accelerated Gradient (NAG), Adagrad, Adadelta, RMSprop, Adam, AdaMax, Nadam, AMSGrad.

1.4. Data Augmentation

When selecting a model, it is crucial to choose one that not only performs quickly but also adequately fits the data structure. Insufficient or excessive fitting of the model to the data can significantly impact predictions [33]. To avoid issues like underfitting and overfitting, a balance between bias and variance must be found. To address overfitting, more data can be collected, or if additional data collection is not feasible, the existing training set can be augmented using data augmentation techniques. In image processing, data augmentation techniques include mirroring, flipping, scaling, adding noise, blurring, and cropping existing images. However, it is essential to avoid transforming the training set into a data dump by adding unnecessary data. Besides data augmentation, selecting a simpler model instead of a complex one, applying regularization, or analyzing the model's performance on the validation set and applying early stopping to the algorithm are viable strategies.

1.5. Hyperparameters

In machine learning, model inputs are called parameters, and hyperparameters are settings adjusted during optimization to train the model better and faster. Any configuration setting that can affect performance is a hyperparameter [33]. Many deep learning algorithms have specific hyperparameters that control the behavior of the algorithm. These hyperparameters affect not only the training time and memory cost but also the quality of the model, i.e., how well the model is trained on the training set and how accurately it predicts new inputs [29].

1.6. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are deep learning models specifically designed to process image data and extract features. These models consist of layers capable of recognizing and classifying patterns in a dataset. In each layer, filters are applied to the input data to create feature maps, highlighting important features. These feature maps are then combined to recognize more complex features, ultimately leading to classification [38-39]. Advancements in neural network architectures have significantly progressed, especially with the success of the AlexNet architecture in achieving remarkable performance on image data. Improvements in computational power have enabled the development of more complex models. This study focuses on classifying laboratory images obtained from comet assay experiments into five different levels of damage using a CNN architecture. Various models' performances are evaluated based on their success in classifying the damage levels.

2. Materials and Methods

In this study, peripheral blood mononuclear cell images were used for conducting the comet assay. The preparation of images followed the steps schematically illustrated in Figure 3, adhering to the comet assay methodology. The comet assay, or single-cell gel electrophoresis, is a technique used to detect DNA damage following the interaction of cells with harmful substances. This sensitive, reliable, and rapid method can identify DNA damage at the cellular level. It is widely used to measure DNA damage levels, which are indicators of genotoxic and cytotoxic effects. The method involves isolating cells and embedding them in agarose gel on a slide. These cells are then subjected to an electrophoretic field, which causes DNA fragments of varying charge and molecular weight to migrate based on the extent of damage. These DNA fragments are subsequently stained with a DNA-specific fluorescent dye and evaluated under a fluorescence microscope. The comet assay is effective in accurately, precisely, and quickly measuring single or double-strand breaks in DNA caused by various genotoxic agents such as oxidative stress, toxic heavy metals, chemical agents, drugs, and ultraviolet radiation, using minimal sample volume [8-10].

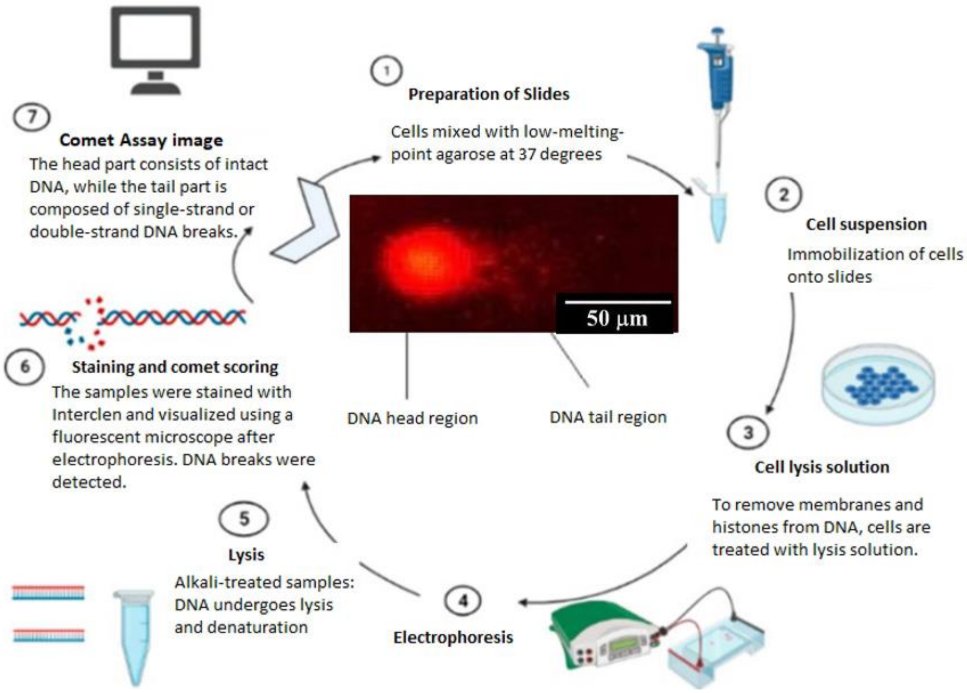

1
Preparation of Slides
Cells mixed with low-melting-point agarose at 37 degrees
2
Cell suspension
Immobilization of cells onto slides
3
Cell lysis solution
To remove membranes and histones from DNA, cells are treated with lysis solution.
4
Electrophoresis
5
Lysis
Alkali-treated samples: DNA undergoes lysis and denaturation
6
Staining and comet scoring
The samples were stained with Interclen and visualized using a fluorescent microscope after electrophoresis. DNA breaks were detected.
7
Comet Assay image
The head part consists of intact DNA, while the tail part is composed of single-strand or double-strand DNA breaks.
50 μm
DNA head region
DNA tail region

Figure 3. Schematic Representation of Comet Assay Analysis Methodology

Dataset

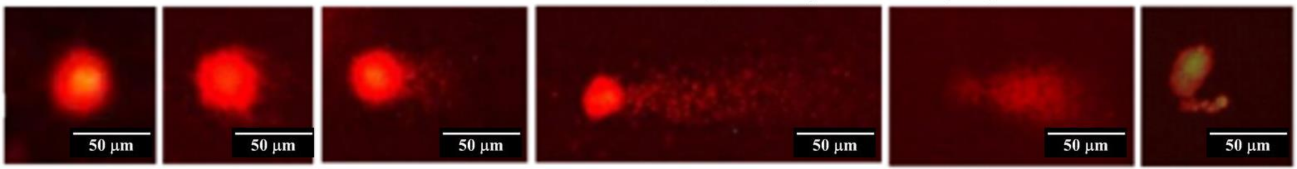
After completing the comet assay analysis, the next step is the labeling process, which must be performed by experienced researchers with great attention to detail. Various methods have been studied to determine the extent of DNA damage in comet assay studies. In some studies, DNA damage is measured by comparing the ratio of damaged cells, which show migration, to intact cells. While this method provides information on overall DNA damage, it does not offer insights into DNA damage at the cellular level. Other studies have measured DNA damage at the cellular level by scoring cells based on their migration rates. DNA migration is typically expressed in micrometers (μm). The length of migration is related to the fragmentation rate and, consequently, to DNA damage. Migration length can be measured in several ways: using a micrometer, rulers overlaid on cell photographs, or monitors in imaging systems. The length of the tail alone does not accurately reflect DNA damage; the percentage of DNA dragged in the tail region is also important. This has led to the definition of the term "tail moment," which is described by the following equation [8-9, 40].

$$\text{tail moment} = \text{tail length } x \begin{pmatrix} \text{fluorescence intensity in the tail} \\ or \\ \text{percentage of DNA in the tail} \end{pmatrix} \qquad (1)$$

In this study, the tail analysis images were independently labeled under the guidance of two experienced biology researchers. The open-source software LabelImg was used for manual labeling of the images.

In deep learning studies, data dependency often poses a significant issue. To address this problem, data augmentation is employed, where different characteristics of the same image (such as rotation, skewing, width shift, height shift, zooming, horizontal flip, and vertical flip) are altered to increase the number of images and used in the training process. This approach aims to achieve higher accuracy rates during training. Utilizing more data can lead to greater success [41-43]. The dataset is divided into 80% for training and 20% for testing.

The labeled images consist of those with noisy backgrounds and those with clean backgrounds. All images were included in the labeling process for training purposes. It was considered beneficial to use the obtained images in the study, as images captured by expert researchers may not always be clean. However, it was particularly noted that including all images in the training set could pose some challenges in classification (Figure 4).



2.1. Deep Learning Methodology

Figure 4. Comet analysis was performed on cells using fluorescence microscopy to evaluate DNA damage. The extent of damage was classified into five distinct categories. (0) No damage: Characterized by a circular image with no tail, indicating a robust structure (C0). (1) Low damage: The tail length is up to twice the diameter of the nucleus (C1). (2) Moderate damage: The tail length ranges from two to three times the diameter of the nucleus (C2). (3) High damage: The tail length is approximately three times the diameter of the nucleus (C3). (4) Severe damage: The tail length exceeds three times the diameter of the nucleus (C4). Additionally, any objects outside the nuclear structure were identified as errors (C6).

In this study, a specialized model was developed for the comet assay segmentation task and compared with some basic models. The model was implemented using the open-source TensorFlow library. The study consists of several distinct sections. The main structure can be divided into feature extraction, classification, and bounding box sections. As shown in Figure 5, the process begins by taking the images as input into the program. The labeled sections are then individually separated into training, testing, and validation datasets. Feature maps are generated, and a scanning process is conducted to identify the regions most likely to contain the objects. Subsequently, a classifier refines the position by creating bounding boxes around the detected objects. In the model structure, the convolutional layer is the most important component of the CNN. The neural structures in the first convolutional layer are not connected to every pixel in the input image. They are only connected to the pixels within their receptive fields. However, in the second convolutional layer, each neuron is connected only to the neurons within a small region of the first layer. This architecture allows the first hidden layer to focus on low-level features and then combine these into higher-level features in the subsequent hidden layer. The architecture of the CNN is illustrated as shown in Figure 5.

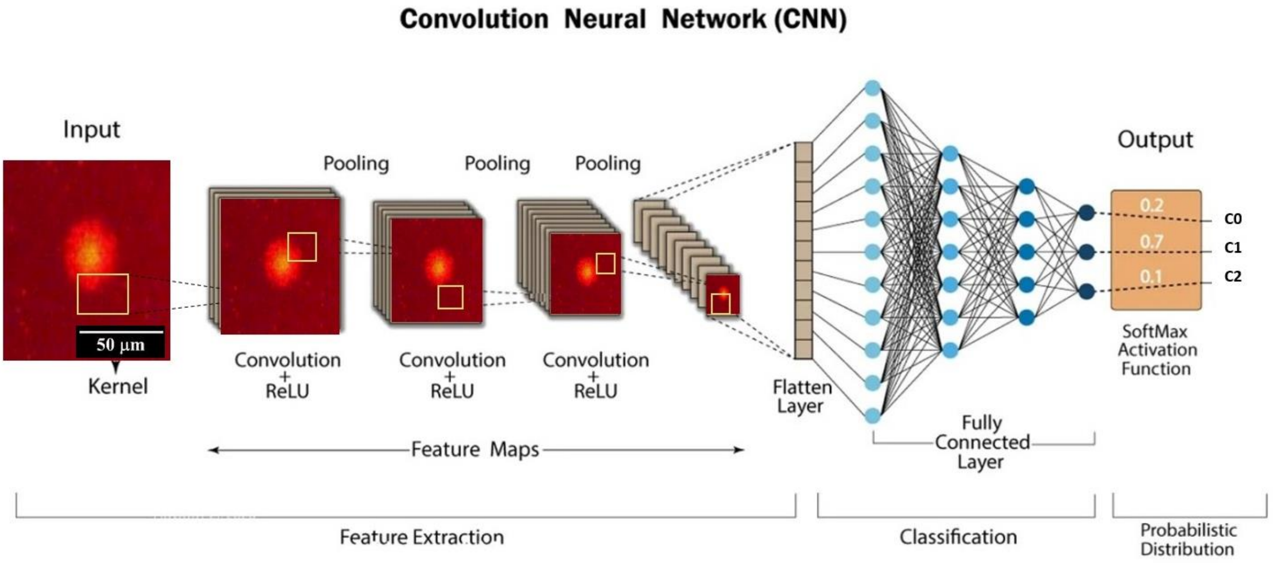


Figure 5. CNN model Structure

In the convolutional layers, multiple trainable filters are applied to the inputs simultaneously, enabling the system to detect multiple features within the inputs. By having all neurons in a feature map share the same parameters, the aim is to reduce the number of model parameters. When the CNN model learns to recognize a pattern in one location, it will be able to recognize similar patterns in different locations more easily. Input images typically have three channels (R, G, B: Red, Green, Blue) since they are usually colored images. If the image is grayscale, it will be processed as a single channel.

When looking more closely at the processing, in a convolutional layer, a neuron located at a specific row and column in the feature map depends on the outputs of the neurons in the previous layer across all feature maps, considering the width and height strides. The following equation shows how the output of a neuron is calculated:

$$z_{i,j,k} = b_k + \sum_{u=0}^{f_{h^{-1}}} \sum_{v=0}^{f_{w^{-1}}} \sum_{k'=0}^{f_{n'^{-1}}} x_{i',j',k'} \, X \, w_{u,v,k',k} \tag{2}$$

$$i' = i \; x \; s_h + u \tag{3}$$

$$j' = j \; x \; s_w + v \tag{4}$$

In this equation, $\boldsymbol{z_{i,j,k}}$ represents the output of the neuron located at the $i'th$ row and $j'th$ column of the k feature map in the convolutional layer. s_h and s_w denote the vertical and horizontal stride lengths, while f_h and f_w represent the height and width of the receptive field, respectively. $f_{n'}$ indicates the number of feature maps in the previous layer. $\boldsymbol{x_{i',j',k'}}$ represents the output of the neuron located at the $i'th$ row, $j'th$ column, and $k'th$ feature map in the previous layer. b_k is the bias term for the $k'th$ feature map. $w_{u,v,k',k}$ represents the connection weight between a neuron in the $k'th$ feature map of the previous layer and its input located at the $u'th$ row and $v'th$ column of the $k'th$ feature map. In TensorFlow, each input image is represented as a three-dimensional tensor with dimensions corresponding to height, width, and channels. The convolutional layer weights, when considering batch size, are represented as four-dimensional tensors. In the max-pooling or pooling layer, aggregation is performed, reducing memory usage and the number of parameters while retaining the maximum values. Typical CNN architectures are built by stacking a few convolutional layers, followed by a pooling layer, then additional convolutional layers, and another pooling layer. This structure is repeated, leading to a gradual reduction in the spatial dimensions of the image as it progresses through the network, while the depth of the network increases. At the top of this stack, fully connected layers are added, forming a feedforward network, which ultimately outputs a prediction value in the final layer. When constructing the model, convolution kernels are often chosen to be large. As the kernel size decreases, the model uses fewer parameters and requires less computation, potentially resulting in better performance. In the model's operation, the first convolutional layer may use a large kernel size with a stride of two or more, aiming to reduce the spatial dimensions without losing much visual information. The input images for the model's training were processed at resolutions of 224x224, 180x180, 196x196, and 128x128 pixels.

2.2. Performance Metrics

Performance metrics were determined using the confusion matrix of the developed models. The goal is to identify, for example, how many images from the C0 class were misclassified into the C1 class. In the confusion matrix, each row represents the actual class, while each column represents the predicted class value.

- **True Positive (TP):** The case where a condition that is generally positive is correctly predicted as positive.
- **True Negative (TN):** The case where a condition that is negative is correctly predicted as negative.
- **False Positive (FP):** The case where a condition that is negative is incorrectly predicted as positive.
- **False Negative (FN):** The case where a condition that is positive is incorrectly predicted as negative.

The confusion matrix provides a wealth of information, including accuracy, precision, recall, and F1 score, which are critical for evaluating the performance of the study's results. Given the imbalanced class distribution in the dataset, the F1 score is particularly important [45 - 47]. The F1 score is calculated as the harmonic mean of precision and recall, providing a single metric that balances the two. The equations for these metrics are as follows:

$$Accuracy = \frac{TP+TN}{(TP+TN+FP+FN)} \tag{5}$$

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

$$Recall(Sensitivity) = \frac{TP}{TP+FN} \tag{7}$$

$$F1\ Score = 2 * \frac{Precision\ x\ Recall}{Precision + Recall} \tag{8}$$

Considering the imbalanced numerical distribution of the dataset, focusing on the F1 score is crucial for a comprehensive evaluation of the model's performance.

2.3. Convolutional Neural Network Architecture

In this study, different models were developed to identify the architecture that yields the highest performance. The best-performing model was then compared and further developed using pre-defined existing model architectures. The models, defined in Table 1 below, were prepared and trained using the images.

Table 1. Hyperparameter values

HiperParameter	Values
CNN Layers	4, 5, 6, 7
Filter Size	6, 16, 32, 64, 128, 256
Pixel Size	128, 180, 196, 224
Kernel Size	3, 5, 7
Dense Layer	1, 2, 3, 4, 5
Epoch	50, 100, 200
Batch Size	16, 32, 128
Optimizer	Adam, SGD, Nadam, Adamax

Hyperparameter tuning is a critical aspect of designing an effective convolutional neural network (CNN) architecture. CNNs have shown remarkable success in various computer vision tasks such as image classification, object detection, and segmentation. However, their performance largely depends on the choice of hyperparameters. This study highlights the importance of hyperparameter tuning in CNNs. Typically, CNNs include several hyperparameters that significantly affect their performance, such as "learning rate," "batch size," "number of filters," "kernel size," and "dropout rate." Proper hyperparameter tuning is crucial to ensure the model's convergence, generalization, and efficiency. Additionally, selecting an appropriate learning rate is vital to prevent slow convergence or overshooting the target. The batch size determines the number of examples used in each training iteration. The number of filters and kernel size in the convolutional layers affect the model's capacity to learn hierarchical features from the input data. Choosing the right number and size of filters is crucial for effective feature extraction. Dropout is a parameter that helps prevent overfitting by randomly dropping a portion of neuron outputs during training. An appropriate dropout rate is necessary to balance model complexity and generalization. Hyperparameter tuning can be time-consuming and computationally expensive, especially for large datasets and complex CNN architectures. Improper hyperparameter settings can lead to overfitting, where the model performs well on training data but fails to generalize to unseen data. Proper hyperparameter selection can significantly impact the model's convergence, generalization, and efficiency [37].

The hyperparameters in Table 1 define the specific configurations for each model, such as the number of CNN layers, number of filters, input image sizes, kernel sizes, dense layers, number of epochs, batch sizes, and optimizers. These models were trained on the dataset to evaluate their performance and identify the most effective architecture for comet assay segmentation. The training process involved inputting the labeled images, extracting features, and optimizing the model parameters to achieve the best classification results. The CNN model structure was carefully designed and optimized due to the close boundaries resulting from the similarities between classes. Considering all the hyperparameter values in Table 1, there are a total of 4320 different CNN network combinations. Training and comparing all these combinations would require significant time. Consequently, 50 models were trained, and the results of 10 models were tabulated [48- 49].

In convolutional neural network architectures, increasing the depth (number of convolutional layers) and width (number of filters) of the hyperparameters enhances the learning capacity of the model during training. However, having more network weight parameters in models with limited datasets increases the risk of overfitting. To mitigate this overfitting issue, regularization techniques such as dropout, batch normalization, and data augmentation are employed. During the development phase of the CNN architecture, the impact of changing hyperparameter values on model training was investigated. The goal was to determine the most suitable hyperparameter configuration for the model. Table 2 presents some of the model configurations tested. Using excessively large kernel sizes in model training can reduce the training performance. Smaller kernel sizes result in fewer parameters and require less computation. In convolution operations, the first layer can use larger kernels (such as 5x5 or 7x7), which does not cause significant information loss but reduces the spatial dimensions of the image. To prevent overfitting and improve the quality of training, dropout regularization is added to the model [50 - 52]. The explanations of the data defined in Table-2 are provided. NCL(Neuronal Connectivity Layer), NF (Number of Features), KS (Kernel Size), IID(Pixel size), Bs(Batch Size), Opt(optimization algorithm), Acc(Accuracy).

Table 2. The feature structures of a model

Model No	NCL	NF	KS	IID	Epoch	Bs	Opt	Acc	F1
1	6	6, 16, 32, 64, 128, 256	5, 3, 3, 3, 3,3	224	50	8	SGD	0.84	0.83
2	6	16, 16, 32, 32, 64, 128	5, 3, 3, 5, 5, 5	196	50	8	Adam	0.86	0.85
3	7	16, 16, 32, 32, 64, 128, 128	5, 5, 5, 5, 5, 5, 5	128	100	16	Adam	0.87	0.87
4	7	16, 32, 32, 32, 64, 128, 256	5, 3, 3, 3, 7, 7, 7	128	200	16	SGD	0.85	0.84
5	5	16, 32, 32, 64, 256	5, 3, 3, 5, 5	196	100	16	Adamax	0.85	0.85
6	6	16, 32, 64, 128, 256,512	5, 3, 3, 3, 3,3	196	200	16	Adam	0.82	0.81
7	6	6, 32, 32, 64, 128, 256	5, 3, 3, 3, 3,3	128	100	16	Adamax	0.88	0.88
8	6	16, 32, 32, 128, 128	5, 3, 3, 3, 7, 7	224	100	16	Adam	0.89	0.88
9	7	16, 32, 32, 128, 128, 256	5, 3, 3, 3, 3,3	224	100	8	Adam	0.9	0.9
10	5	16, 32, 128, 256, 512	3, 3, 3, 3, 3	196	200	8	Adamax	0.92	0.92

As a result of the studies conducted, the performance levels were obtained as shown in Table 2. The accuracy rates and F1 scores of the developed models are presented in Figure 6.

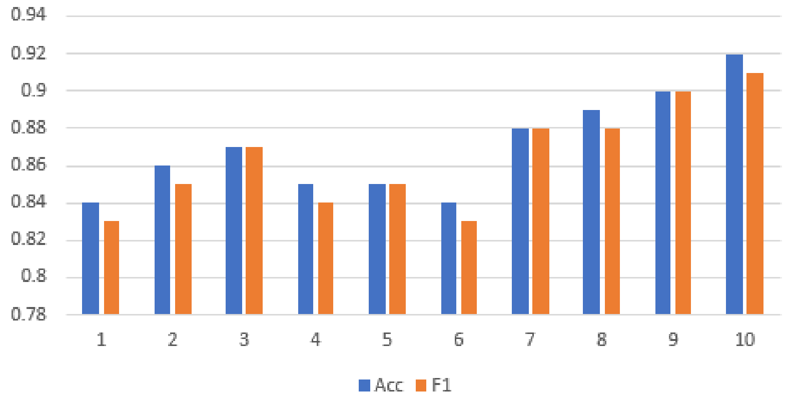


Figure 6. Custom models Accuracy and F1 Scores

In the specified 10 models, the convolutional layers were set to 5, 6, and 7, with the number of filters varying between models, including values such as 6, 16, 32, 64, 128, 256, and 512. The number of epochs was set to either 50 or 100. Additionally, in studies using transfer learning, the accuracy values and F1 scores of the models are shown in Figure 7. After the model trainings were completed, the results were evaluated, and the confusion matrix shown in Figure 8 was prepared.

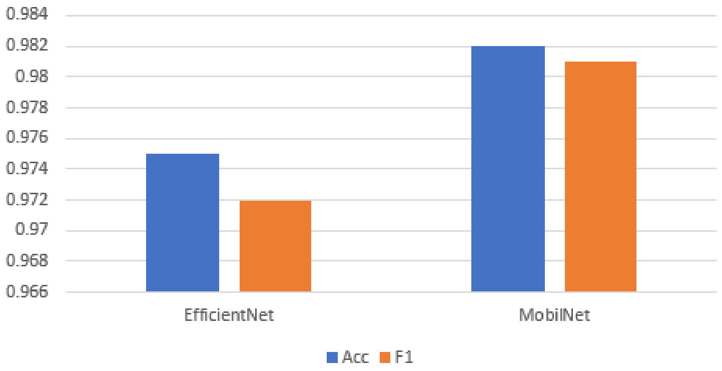
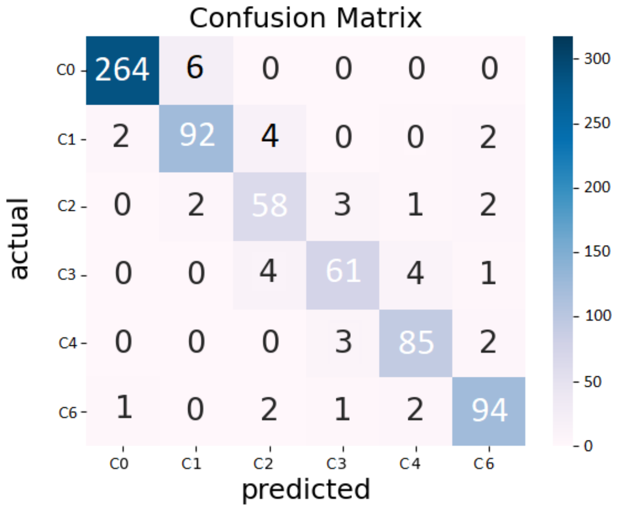


Figure 7. Transfer Learning EfficientNet model and MobilNet model Accuracy and F1 scores



	C0	C1	C2	C3	C4	C6
C0	264	6	0	0	0	0
C1	2	92	4	0	0	2
C2	0	2	58	3	1	2
C3	0	0	4	61	4	1
C4	0	0	0	3	85	2
C6	1	0	2	1	2	94

Figure 8. Confusion Matrix

Based on the DNA damage values, the classes were defined as C0, C1, C2, C3, C4, and C6, as previously mentioned. Predictions were made on the images in the test dataset. Out of 270 images with a C0 damage level, 261 were correctly predicted as C0 (True Positive). Due to similarities with the C1 class, 9 images were incorrectly classified as C1 (False Negative).
Similarly, when examining the prediction values for other classes, it was generally observed that the errors occurred in the classes adjacent to the class being analyzed, due to high similarities. For C0 class predictions, (2+1+3) images were classified as false positives for adjacent classes. The calculations were performed based on these values using equations (4, 5, 6, and 7).

3. Results

Comet images were classified into five classes (C0, C1, C2, C3, C4, C6) with approximately 92% accuracy. The success of the classification primarily depends on accurately labeling the data by determining the most precise distinctions. To further increase the success rate, it is necessary to use images with more distinctive features that can differentiate the classes, which means improving image quality is essential. One of the key observations in this study is that distinguishing between C0 and C3 images is easier, whereas distinguishing between C0 and C1 images is more challenging. This observation is evident from the confusion matrix. A detailed examination of the results shows that the predicted classes are very close to the actual classes, to the extent that even an expert might fail in classifying the images. Similarities between images in neighboring classes directly affect the success rate. Therefore, having more specific data is critical to improving classification success rates. In this study, techniques such as group normalization and dropout were used to prevent overfitting. The ReLU activation function was preferred based on the results. The Adam optimizer was used as the optimization function. Models were developed with different network layers, trained for 50, 100, and 200 epochs. A table of randomly selected models from these was created, as shown in Table 2. Overall, the aim of this study was to develop a model with a high accuracy rate capable of effectively classifying comet images. However, improvements can be made by increasing the sample size and using more specific features for training. After training and testing, the performance of the prepared models was compared across different model combinations. The accuracy results of the prepared network models and models trained with the transfer learning method are shown in Figures 7 and 8.

4. Conclusion and Future Works

In the continuation of the study, the developed model will be used as the backbone in the Faster-RCNN architecture, which demonstrates very good performance in object recognition, and will be employed for the detection and classification of real DNA damage images. The obtained DNA damage images will be categorized into predefined classes based on the extent of damage. This will increase the number of DNA damage images using real images. Depending on the increase in the number of images, retraining can be performed at specific intervals, and the model's performance will be enhanced to higher levels.

5. References

[1] Hoeijmakers, J. H. DNA damage, aging, and cancer. N. Engl. J. Med. 361, 1475–1485 (2009).

[2] Kadioglu, E., Sardas, S., Aslan, S., Isik, E. & Karakaya, A. E. Detection of oxidative DNA damage in lymphocytes of patients with Alzheimer’s disease. Biomarkers 9, 203–209 (2004).

[3] Kopjar, N., Garaj-Vrhovac, V. & Milas, I. Assessment of chemotherapy-induced DNA damage in peripheral blood leukocytes of cancer patients using the alkaline comet assay. Teratog. Carcinog. Mutagen. 22, 13–30 (2002).

[4] Collins, A. R. et al. DNA damage in diabetes: Correlation with a clinical marker. Free Radical Biol. Med. 25, 373–377 (1998).

[5] A.R. Collins, M. Ai-Guo, S.J. Duthie, The kinetics of repair of oxidative dna dam-age (strand breaks and oxidised pyrimidines) in human cells, Mutat. Res./DNA Repair 336 (1) (1995) 69–77.

[6] D.W. Fairbairn, P.L. Olive, K.L. O’Neill, The comet assay: a comprehensive review, Mutat. Res./Rev. Genet. Toxicol. 339 (1) (1995) 37–59.

[7] M. Kuchařová, M. Hronek, K. Rybáková, Z. Zadák, R. Štětina, V. Josková, A. Patková, Comet assay and its use for evaluating oxidative dna damage in some pathological states, Physiol. Res. 68 (1) (2019) 1–15.

[8] Ostling, O. & Johanson, K. J. Microelectrophoretic study of radiation-induced DNA damages in individual mammalian cells. Biochem. Biophys. Res. Commun. 123, 291–298 (1984).

[9] Singh, N. P., McCoy, M. T., Tice, R. R. & Schneider, E. L. A simple technique for quantitation of low levels of DNA damage in individual cells. Exp. Cell Res. 175, 184–191 (1988).

[10] Helma, C. & Uhl, M. A public domain image-analysis program for the single-cell gel-electrophoresis (comet) assay. Mutat. Res. Genet. Toxicol. Environ. Mutagenesis 466, 9–15 (2000).

[11] Gyori, B. M., Venkatachalam, G., Thiagarajan, P., Hsu, D. & Clement, M.-V. OpenComet: An automated tool for comet assay image analysis. Redox Biol. 2, 457–465 (2014).

[12] N. Van Eck, L. Waltman, Software survey: VOSviewer, a computer program for bibliometric mapping, Scientometrics 84 (2) (2010) 523–538, https://doi.org/ 10.1007/s11192-009-0146-3.

[13] S. Alonso, F.J. Cabrerizo, E. Herrera-Viedma, F. Herrera, h-Index: a review focused in its variants, computation and standardization for different scientific fields, J. Infor. 3 (4) (2009) 273–289, https://doi.org/10.1016/j.joi.2009.04.001.

[14] M. Laakso, A. Klippi, A closer look at the'hint and guess' sequences in aphasic conversation, Aphasiology 13 (4–5) (1999) 345–363, https://doi.org/10.1080/ 026870399402136.

[15] D.J. Clarke, Nursing practice in stroke rehabilitation: systematic review and meta-ethnography, J. Clin. Nurs. 23 (9–10) (2014) 1201–1226, https://doi.org/ 10.3109/07434618.2014.955614.

[16] G. Chen, L. Xiao, Selecting publication keywords for domain analysis in bibliometrics: a comparison of three methods, J. Infor. 10 (1) (2016) 212–223, https:// doi.org/10.1016/j.joi.2016.01.006.

[17] Sreelatha, G., Muraleedharan, A., Sathidevi, P. S., Chand, P., and Rajkumar, R. P., "CometQ: An automated tool for the detection and quantification of DNA damage using comet assay image analysis", Computer Programs and Methods in Biomedicine, 133: 143-154 (2016)

[18] Lee, T., Lee, S., Sim, W. Y., Jung, Y. M., Han, S., Chung, C., Chang, J. J., Min, H., and Yoon, S., "Robust classification of DNA damage patterns in single cell gel electrophoresis", 35th Annual International Conference of the IEEE EMBS, Osaka, 3666-3669 (2013).

[19] Sreelatha, G., Rashmi, P., Sathidevi, P. S., Aparma, M., Chand, P., and Rajkumar, R. P., "Automatic detection of comets in silver stained comet assay images for dna damage analysis", 2014 IEEE International Conference on Signal Processing,

[20] Sansone, M., Zeni, O., and Esposito, G., "Automated segmentation of comet assay images using Gaussian filtering and fuzzy clustering", Med Biol Eng Comput, 50 (5): 523-532 (2012).

[21] Böcker, W., Rolf, W., Bauch, T., Müller, W., U., and Streffer, C., "Automated comet assay analysis", Cytometry, 35 (2): 134-144 (1999).

[22] Gonzalez, J. E., Romero, I., Barquinero J. E., and Garcia, O., "Automatic analysis of silver-stained comets by CellProfiler software", Mutation Research, 748 (1): 60-64 (2012).

[23] Riccardo Rosati, Luca Romeo, Sonia Silvestri, Fabio Marcheggiani, Luca Tiano, Emanuele Frontoni "Faster R-CNN approach for detection and quantification of DNA damage in comet assay images" Computers in Biology and Medicine 123 (2020) 103912

[24] Attila Beleon, Sara Pignatta, Chiara Arienti, Antonella Carbonaro, Peter Horvath, Giovanni Martinelli, Gastone Castellani, Anna Tesei, Filippo Piccinini "CometAnalyser: A user-friendly, open-source deep-learning microscopy tool for quantitative comet assay analysis", Computational and Structural Biotechnology Journal 20 (2022) 4122–4130

[25] Srikanth Namuduria, Prateek Mehta, Lise Barbe, Stephanie Lam, Zohreh Faghihmonzavi, Steve Finkbeiner, Shekhar Bhansali "Faster Deep Ensemble Averaging for Quantification of DNA Damage from Comet Assay Images With Uncertainty Estimates", arXiv:2112.12839v1 [q-bio.QM] 23 Dec 2021

[26] Afiahayati, Edgar Anarossi, Ryna Dwi Yanuaryska and Sri Mulyana, "GamaComet: A Deep Learning-Based Tool for the Detection and Classification of DNA Damage from Buccal Mucosa Comet Assay Images", Diagnostics 2022, 12, 2002. https://doi.org/10.3390/diagnostics12082002

[27] Lewis, N. D. (2016). Deep Learning Made Easy with R. Auscov, ABD

[28] Mousavi, S. S., Schukat, M., & Howley, E. (2016). "Deep reinforcement learning: an overview". In Proceedings of SAI Intelligent Systems Conference: 426-440.

[29] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press, ABD.

[30] Agrawal, A. "Loss functions and optimization algorithms". https://medium.com: https://medium.com/data-science-group-iitr/loss-functions-and-optimizationalgorithms-demystified-bb92daff331c

[31] Sharma, A. "Understanding activation functions in deep learning". https://www.learnopencv.com/understanding-activation-functions-in-deeplearning/

[32] Mehrotra, K., Mohan, C. K., & Ranka, S. (1996). Elements of Artificial Neural Networks. USA: MIT Press.

[33] Patterson, J., & Gibson, A. (2017). Deep Learning: A Practitioner's Approach. O'Reilly Media, Inc, ABD.

[34] Heaton, J. (2015). Artificial Intelligence for Humans, Volume 3: Neural Networks and Deep Learning. Heaton Research Inc, Chesterfield, ABD.

[35] Gulli, A., & Pal, S. (2017). Deep Learning with Keras. . Packt Publishing Ltd, Birmingham.

[36] Maas, A.L., Hannun, A.Y., & Ng, A.Y. (2013). "Rectifier nonlinearities improve neural network acoustic models". In Proc. icml (Vol. 30, No. 1): 3-10.

[37] Almufadi, N., & Qamar, A.M. (2022). Deep Convolutional Neural Network Based Churn Prediction for Telecommunication Industry. Computer Systems Science and Engineering, 43(3), 1255-1270. https://doi.org/10.32604/csse.2022.025029

[38] Ganapathy, S., Muraleedharan, A., Sathidevi, P. S., Chand, P. & Rajkumar, R. P. CometQ: An automated tool for the detection and quantification of DNA damage using comet assay image analysis. Comput. Methods Programs Biomed. 133, 143–154 (2016).

[39] Smochina, C., Manta, V., Kropatsch, W., Crypts detection in microscopic images using hierarchical structures, Pattern Recognit. Lett. 34 (8) (2013) 934–941, Computer Analysis of Images and Patterns.

[40] Rojas E, Lopez MC, Valverde M. Single cell gel electrophoresis assay: methodology and applications. J. Chromatogr, 1999; 722: 225-54.

[41] McKelvey-Martin VJ, Green MH, Schmezer P, Pool-Zobel BL, DeMeo MP, Collins A. The single cell gel electrophoresis assay (comet assay): a European review. Mutat. Res, 1993; 288(1): 47-63.

[42] Agarwal A, Erenpreiss J, Sharma R. Sperm chromatin assessment. In: Gardner DK, Weissman A, Howles CM, Shoham Z. Eds. Textbook of Assisted Reproductive Technologies. 3rd edition, India: Replika Pres Pvt. Ltd, 2009:67-84.

[43] Olive PL, Durand RE, Banath JP, Evans HH. Etoposide sensitivity and topoisomerase II activity in Chinese hamster V79 monolayers and small spheroids. Int. J. Radiat. Biol, 1991; 60(3): 453-66.

[44] https://medium.com/@sebinbusra/evrisimsel-sinir-agi-convolutional-neural-network-f49f7b65c72

[45] Xu, X., Yang, Z., Wang, Y., A method based on rank-ordered filter to detect edges in cellular image, Pattern Recognit. Lett. 30 (6) (2009) 634–640.

[46] Shen, D., Wu, G., Suk, H.I., Deep learning in medical image analysis, Annu. Rev. Biomed. Eng. 19 (Jun (1)) (2017) 221–248.

[47] Schmidt-Richberg A, Brosch T, Schadewaldt N, Klinder T, Caballaro A, Salim I, et al., "Abdomen segmentation in 3D fetal ultrasound using CNN-powered deformable models", (2017)

[48] Kamnitsas K, Ledig C, Newcombe VFJ, Simpson JP, Kane AD, Menon DK, et al. Zheng Y, Barbu A, Georgescu B, Scheuering M, Comaniciu D., "Four-chamber heart modeling and automatic segmentation for 3-D cardiac CT volumes using marginal space learning and steerable features", (2008)

[49] Dietterich, T.G., Approximate statistical tests for comparing supervised classi-fication learning algorithms, Neural Comput. 10 (7) (1998) 1895–1923, http: //dx.doi.org/10.1162/089976698300017197.

[50] Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., The pascal visual object classes (voc) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.

[51] Ren, S., He, K., Girshick, R. & Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems 91–99 (2015).

[52] Atila, Ü., Baydilli, Y. Y., Sehirli, E. & Turan, M. K. Classification of DNA damages on segmented comet assay images using convo-lutional neural network. Comput. Methods Programs Biomed. 186, 105192 (2020).

[53] Azadvatan Y., Kurt M. Azadvatan Y., Kurt M. MelNet: A Real-Time Deep Learning Algorithm for Object Detection, Computer Vision and Pattern Recognition (2024) https://doi.org/10.48550/arXiv.2401.17972