Classification of Human Sperm Based on Morphology Using Densely Connected Convolutional Networks

Aristoteles Aristoteles[1], Admi Syarif[2], Dewi Asiah Shofiana[3], Irma Azizah[4]

[1] Doctoral Program of Mathematics and Natural Sciences, Lampung University
[1,2,3,4] Department of Computer Science, Faculty of Mathematics and Natural Sciences
[1,2,3,4] University of Lampung, Bandar Lampung, Indonesia
aristoteles.1981@fmipa.unila.ac.id*

Abstract. Male infertility, which accounts for about 50% of all infertility cases, is assessed through sperm morphology analysis. However, the complex variation of sperm shapes often complicates the diagnosis process. Automated systems offer a more accurate solution than manual selection. This study aims to implement the DenseNet169 CNN architecture for sperm classification based on morphology and to test its performance with different data sharing schemes. On the HuSHem dataset, the highest accuracy of 97.78% was obtained with a data sharing ratio of 70:25:5, while the SCIAN dataset achieved an accuracy of 78.79% at the same ratio. DenseNet169, with its dense connectivity, is proven to be effective in overcoming the gradient vanishing problem and improving feature efficiency, resulting in high performance in sperm morphology classification.

Keywords: Densenet, Male infertility, Sperm morphology.

1 Introduction

The World Health Organization (WHO) defines infertility as the inability to conceive after at least 12 months of regular, unprotected intercourse. Infertility is estimated to affect 8–12% of couples in the reproductive age group [1], with male factors contributing to approximately 50% of all infertility cases [5]. Evaluating male infertility involves analyzing semen samples in the laboratory, focusing on parameters such as sperm morphology, concentration, and motility [13]. Abnormal sperm morphology, characterized by defects in the shape and size of sperm cells, plays a critical role in male infertility. Therefore, accurately classifying sperm morphology is an essential step in diagnosing male infertility. However, the variations in sperm shape make this analysis complex and challenging, even for specialists [13].

Traditionally, manual sperm selection has been a common practice, but it has limitations in terms of accuracy and efficiency. In contrast, automated systems provide a more efficient solution by reducing the need for skilled technicians and saving labor in sperm selection processes, such as intracytoplasmic sperm injection [6]. Classifying sperm morphology is a complex task due to the diverse forms that are difficult to dis-

S. Hadi et al. (eds.), *Proceedings of the 5th International Conference on Applied Sciences, Mathematics, and Informatics (ICASMI 2024)*, Advances in Physics Research 13,
https://doi.org/10.2991/978-94-6463-730-4_6

tinguish, even for experts. This diversity includes complexities in shape, size, and texture, making the task particularly challenging. Moreover, there is intra-class variation and inter-class similarity, such as elongated amorphous heads resembling tapered heads, or pyriform heads appearing similar to tapered ones [4].

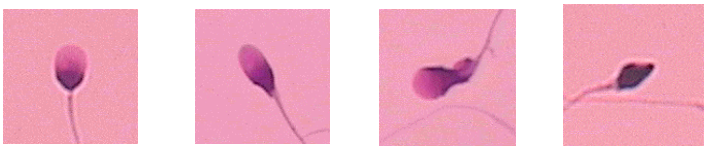
In recent decades, lifestyle factors have been analyzed as potential causes of declining sperm quality in humans, with studies examining aspects such as dietary habits, exposure to harmful chemicals, and physical activity [7]. However, interpreting such data accurately and efficiently can be challenging [7]. This is where deep learning offers a promising solution. As a branch of machine learning, deep learning has produced excellent results in various fields, including non-medical applications [8]. Deep learning allows models to learn feature representations directly from raw data, making it highly effective in diagnostic tasks, such as detecting abnormalities through deep neural networks [12].

Deep learning also holds significant potential for analyzing human sperm morphology. This technology can be applied to extract and map visual features from sperm images to relevant morphological categories [10]. According to experts, deep learning relies on Convolutional Neural Networks (CNN) to predict both sperm motility and morphology [3]. Numerous studies have been published on sperm morphology classification, highlighting the effectiveness of deep learning in providing more accurate and efficient solutions for diagnosing male infertility.

2 Materials and Methodology

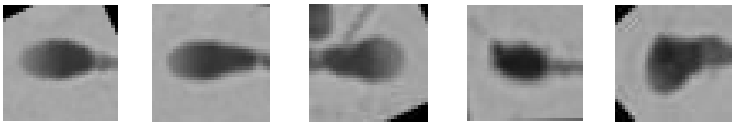
2.1 Dataset

The data used in this research include two datasets: HuSHem and SCIAN (SCIAN Gold-standard for Morphological Sperm Analysis). The HuSHem dataset originates from a study conducted by Shaker (2018) [9] and can be accessed via [Papers with Code](https://paperswithcode.com/dataset/hushem). The SCIAN dataset, on the other hand, comes from a study by Chang et al. (2017) [2]. The HuSHem dataset contains images of human sperm head morphology, categorized into four classes: normal, pyriform, tapered, and amorphous. These images are in BMP format with a resolution of 131×131 pixels, consisting of a total of 216 sperm cell images. The HuSHem dataset consists of 216 sperm cell images with each class consisting of 54 normal, 53 tapered, 57 pyriform, and 52 amorphous. In contrast, the SCIAN dataset contains 1,854 sperm head images, which were classified by three domain experts from Chile according to the World Health Organization (WHO) criteria. The images are in TIF format with a resolution of 35×35 pixels. The SCIAN dataset consists of 1854 sperm cell images with each class consisting of 100 normal, 228 tapered, 76 pyriform, 72 small, and 656 amorphous. Sample images of the dataset can be seen in Figure 1 and Figure 2.



(a) (b) (c) (d)

Fig. 1. Overview of the HuSHem dataset displaying the five morphological catego-ries: (a) Normal, (b) Tapered, (c) Pyriform, and (d) Amorphous.



(a) (b) (c) (d) (e)

Fig. 2. Overview of the SCIAN dataset displaying the five morphological categories: (a) Normal, (b) Tapered, (c) Pyriform, (d) Small, and (e) Amorphous.

2.2 Pre-processing

Pre-processing is a crucial set of steps applied to input data before feeding it into a CNN to prepare the data for more effective processing and improve the model's performance. In this study, the pre-processing stage includes resizing images and normalizing pixel values from the 0–255 range to a 0–1 range. Additionally, data augmentation techniques are applied to increase the number of training samples by manipulating the original data through various geometric transformations. This augmentation allows the model to learn more generalized patterns and capture essential characteristics of the images, thereby enhancing the model's performance and generalization in image recognition tasks.

In addition to augmentation, cropping is another pre-processing technique applied in this study. Cropping focuses the model's analysis on relevant areas of the image by removing unnecessary elements. Cropping helps eliminate unimportant backgrounds, enabling the model to better capture key details, such as the main object or specific features, thus improving accuracy in classification tasks. In this study, images from the HuSHem dataset were cropped to a size of 105x105 pixels to maximize the important information. However, cropping was not applied to the SCIAN dataset, as its images are smaller, with a resolution of 35x35 pixels. Applying cropping to SCIAN images would risk losing crucial elements, reducing the useful information for analysis or model training. An example of image cropping for the HuSHem dataset is shown in Figure 3.

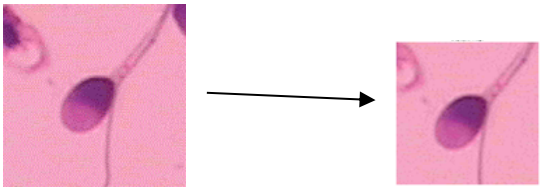


Fig. 3. Image Cropping.

2.3 Data-Splitting Scenarios

The HusHem and SCIAN datasets were each divided into three data-splitting scenarios. This division was performed by first separating the data into training and testing sets, followed by splitting the training data into training and validation sets. The details of the data-splitting scenarios in this study are as follows:

a. The first scenario involves 80% training data, 10% validation data, and 10% testing data.
b. The second scenario involves 75% training data, 25% validation data, and 5% testing data.
c. The third scenario involves 60% training data, 20% validation data, and 20% testing data.

Each of these scenarios was designed to assess the model's performance under different proportions of training, validation, and testing data. By varying the size of the validation and testing sets, the study aims to determine the optimal data-splitting strategy that provides the best balance between model training and evaluation. These scenarios allow for a comprehensive evaluation of the model's ability to generalize to unseen data while ensuring that enough data is available for effective training and validation.

2.4 Convolutional Neural Networks (CNN) Architecture

The model training was conducted on Google Colab for program execution. The training process of the Convolutional Neural Network (CNN) involves several steps to optimize the model's performance in recognizing patterns within image data. First, the prepared image data, which has been divided into training, validation, and testing sets, is fed into the model. This data is processed through several convolutional layers, where automatically learned filters or kernels identify key features of the images, such as edges, textures, or specific patterns. The convolutional layers are followed by pooling layers, which reduce the dimensionality of the features to lower computational load and prevent overfitting, as well as batch normalization layers that help stabilize the training process.

In the DenseNet (Dense Convolutional Network) model, each layer is densely connected to all previous layers, allowing for more efficient feature reuse and reducing the number of parameters. The output from the convolutional and pooling layers is then flattened and passed to fully connected layers, where the model combines the learned

features to make final predictions. The training process itself involves using an optimization algorithm, such as Adam, which updates the model's weights based on the error or loss calculated using a loss function like sparse categorical entropy. Sparse categorical entropy is used for multi-class classification problems where the target labels are encoded as integers. During training, the model is optimized to minimize the loss through backpropagation, where gradients of the loss function are computed and used to update the model's weights.

Additionally, regularization techniques such as dropout and L2 regularization are applied to reduce overfitting and improve the model's generalization ability. L2 Regularization is a type of regularization that penalizes weights proportionally to the sum of the squares of the weights. L2 Regularization helps reduce extreme weights (those with high positive or low negative values) so that they approach 0, but not completely to 0. Features with values very close to 0 remain in the model, but do not greatly affect the model's predictions. L2 Regularization is defined as follows.

$$\text{L2 Regularization} = (w1)2 + (w2)2 + (w3)2 + \ldots (wn)2 \quad (1)$$

Meanwhile, dropout randomly "drops out" some neurons during training, preventing the model from becoming too reliant on specific neurons and improving its ability to generalize. Meanwhile, L2 regularization, or ridge regularization, adds a penalty to large weights in the loss function, which reduces the values of overly large weights. This helps prevent the model from becoming too specific and overfitting by shrinking those weights during training. Overfitting occurs when a machine learning model performs very well on the training data but fails to make accurate predictions on unseen data. By penalizing large weights, the model is encouraged to use smaller, simpler weights, which helps it generalize better to new data.

During training, the model's performance is periodically validated using validation data to monitor its progress. Once the training is complete, the model is tested on unseen test data to evaluate its performance in real-world scenarios. This process ensures that the CNN model is not only well-adapted to the training data but also capable of delivering accurate results on new, unseen data. The following section presents the results of the model training.

2.5 Evaluation

Model evaluation aims to measure the performance and ability of the model in learning feature representations from the data. This evaluation is conducted by calculating evaluation metrics such as accuracy, precision, recall, and F1-score. It provides insights into how well the Convolutional Neural Network (CNN) model generalizes to new data. The results of the evaluation assess the extent to which the model can handle different scenarios and determine whether further adjustments or fine-tuning are necessary to improve its performance. Equation for evaluation metrics:

2. Accuracy
   Accuracy is a measure of how many correct predictions the model makes across the entire test dataset. The equation for accuracy is shown in Equation (2).

$$Accuracy = \frac{\text{TP+TN}}{\text{TP+FF+TN+FN}} \quad (2)$$

3. Precision
   Precision measures how relevant the results provided by the system are, compared to all the results retrieved [11]. The equation for precision is shown in Equation (3).

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

4. Recall (Sensitivity)
   Recall, or sensitivity, is defined as the true positive rate or the accuracy of positive predictions [14]. It measures how many truly relevant results are returned [11]. Recall assesses how many actual positive cases the model successfully predicts. The equation for recall is shown in Equation (4).

$$Recall = TP/(TP + FN) \quad (4)$$

5. F1-Score
   The F1-Score reaches its maximum value when precision equals recall. This indicates that the model has a good balance between correctly identifying positive instances (precision) and finding all relevant positive instances (recall). In other words, when precision and recall are balanced, the F1-Score reflects that the model performs well across both aspects. The equation for the F1-Score is shown in Equation (5).

$$F1 - Score = \frac{2}{\frac{1}{Recall}+\frac{1}{Precision}} = 2\,x\frac{Recall\ x\ Precision}{Recall+Precision} \quad (5)$$

3 Experiment Results

3.1 Result

The training process was carried out 10 times with 100 epochs each. After the model training process is complete, the model accuracy is measured to assess how well the model recognizes or classifies data. This accuracy is obtained from evaluating the model using test data, which is not used during training. The results of this training are obtained based on the hyperparameters based on Table 1.

Table 1. Training Model Result.

Data		Matrix	Data Splitting Ratio		
			80:10:10	70:25:5	60:20:20
HuSHem	Augmentation Applied to All Data	Accuracy	97,73%	97,78%	96%
		Precision	98%	98%	96%
		Recall	98%	98%	96%
		F1 Score	98%	98%	96%
	Augmentation Applied Only to the Training Data	Accuracy	95,83%	100%	82,22%
		Precision	96,5%	100%	85%
		Recall	95,75%	100%	82%
		F1 Score	95,75%	100%	82%
SCIAN	Augmentation Applied to All Data	Accuracy	75,15%	78,79%	73,64%
		Precision	76%	80%	74%
		Recall	75%	79%	74%
		F1 Score	75%	79%	74%
	Augmentation Applied Only to the Training Data	Accuracy	65,22%	60,34%	58,52%
		Precision	64%	59%	55%
		Recall	65%	50%	59%
		F1 Score	60%	57%	56%

Following the comprehensive augmentation of the training data, which involved executing the model over 100 epochs and conducting ten distinct runs, the resulting performance metrics displayed considerable enhancements. This process not only allowed the model to learn from a diverse set of data but also provided a more robust framework for evaluating its performance. As depicted in Table 1, the HuSHem dataset achieved an outstanding accuracy rate of 100% when employing a data splitting scheme with a ratio of 70:25:5. This remarkable outcome indicates that the model was able to accurately predict all test data, thereby demonstrating its robustness, reliability, and effectiveness in handling this particular dataset. Such a high level of accuracy indicates that the model successfully identified and learned the intricate features present within the HuSHem dataset, allowing it to make precise classifications.

Conversely, the performance on the SCIAN dataset revealed a more complex situation. The data splitting scheme that yielded the highest accuracy for this dataset was an 80:10:10 ratio, which resulted in a maximum accuracy of only 65.22%. While this result does indicate some degree of improvement in the model's performance, it is relatively low compared to the nearly perfect accuracy achieved with the HuSHem dataset. This disparity can largely be attributed to the uneven distribution of data within the validation and testing sets of the SCIAN dataset, which likely hindered the model's ability to learn effectively from the available data. The imbalanced distribution may have resulted

in insufficient exposure to critical features, ultimately impacting the model's performance.

Recognizing the limitations presented by the SCIAN dataset, a series of additional experiments were initiated to increase the quantity of data. This strategic approach aims to enrich the dataset's informational content, thereby providing the model with a more comprehensive understanding of the underlying patterns present in the data. The objective was not only to improve the model's performance but also to improve its accuracy further. After executing ten runs with the newly augmented dataset, the findings reported in Table 1 revealed that for the HuSHem dataset, the same data splitting ratio of 70:25:5 led to a slightly lower yet still impressive accuracy of 97.78%. This result confirms the model's consistent performance across multiple runs, reinforcing the validity of its training.

Meanwhile, for the SCIAN dataset, the highest accuracy recorded was 78.89%, achieved with the same 70:25:5 ratio. These results indicate a significant positive impact on the model's performance across both datasets, attributable to the increased data volume. With a larger and more diverse dataset available for training, the model was better equipped to identify and learn complex patterns, thus improving its ability to generalize findings more effectively during the classification of test data. The enhanced accuracy in both datasets suggests that the model can now capture a broader range of features and make more informed predictions.

The strategic increase in data volume for both datasets played a pivotal role in significantly enhancing the model's overall accuracy and performance. The HuSHem model approaches near-perfect accuracy, underscoring its capability in complex classification tasks, while the SCIAN model demonstrated substantial improvement, indicating progress despite the need for further refinements. The introduction of a larger and more diverse dataset has proven crucial, as it empowers the model to classify data with greater precision and efficacy. These findings highlight not only the importance of data augmentation in machine learning tasks but also the potential for ongoing research and experimentation to drive further improvements. Moving forward, it will be important to continue exploring various augmentation techniques and data balancing strategies to ensure the model's effectiveness across different datasets and real-world applications.

4 Conclusions

In summary, the HusHem dataset allowed DenseNet to achieve impressive results, thanks to its balanced class distribution and higher image resolution, while the SCIAN dataset posed more challenges due to its smaller image size and class imbalance. Data augmentation played a critical role in enhancing the model's performance, especially for the minority classes. The DenseNet architecture, with its feature reuse across layers, proved effective in preventing overfitting and ensuring generalization, but further improvements could be explored by incorporating ensemble models or employing more advanced augmentation techniques. These findings underscore the importance of dataset quality and balance in achieving high-performance results in sperm morphology

classification, as well as the potential for deep learning models like DenseNet to contribute to automated reproductive health diagnostics.

References

1. Agarwal, A., Baskaran, S., Parekh, N., Cho, C. L., Henkel, R., Vij, S., Arafa, M., Panner Selvam, M. K., Shah, R.: Male infertility. The Lancet **397**(10271), 319–333 (2021). https://doi.org/10.1016/S0140-6736(20)32667-2
2. Chang, V., Garcia, A., Hitschfeld, N., Härtel, S.: Gold-standard for computer-assisted morphological sperm analysis. Computers in Biology and Medicine **83**, 143–150 (2017). https://doi.org/10.1016/j.compbiomed.2017.03.004
3. Haugen, T. B., Hicks, S. A., Andersen, J. M., Witczak, O., Hammer, H. L., Borgli, R., Halvorsen, P., Riegler, M.: Using Deep Learning to Predict Motility and Morphology of Human Sperm. In: Proceedings of the 10th ACM Multimedia Systems Conference, MMSys 2019, pp. 261–266 (2019). https://doi.org/10.1145/3304109.3325814
4. Iqbal, I., Mustafa, G., Ma, J.: Deep Learning-Based Morphological Classification of Human Sperm Heads. Diagnostics **10**(5), 325 (2020). https://doi.org/10.3390/diagnostics10050325
5. Leslie, S. W., Soon-Sutton, T. L., Khan, M. A.: Male Infertility (2024)
6. Mashaal, A. A., Eldosoky, M. A. A., Mahdy, L. N., Ezzat, K. A.: Classification of Human Sperms using ResNet-50 Deep Neural Network. International Journal of Advanced Computer Science and Applications **14**(2), (2023). https://doi.org/10.14569/IJACSA.2023.0140282
7. Naseem, S., Mahmood, T., Saba, T., Alamri, F. S., Bahaj, S. A. O., Ateeq, H., Farooq, U.: DeepFert: An Intelligent Fertility Rate Prediction Approach for Men Based on Deep Learning Neural Networks. IEEE Access 11, 75006–75022 (2023). https://doi.org/10.1109/ACCESS.2023.3290554
8. Reddy, A. S. B., Juliet, D. S.: Transfer Learning with ResNet-50 for Malaria Cell-Image Classification. In: 2019 International Conference on Communication and Signal Processing (ICCSP), pp. 0945–0949 (2019). https://doi.org/10.1109/ICCSP.2019.8697909
9. Shaker, F.: Human Sperm Head Morphology dataset (HuSHeM) (2018)
10. Spencer, L., Fernando, J., Akbaridoust, F., Ackermann, K., Nosrati, R.: Ensembled Deep Learning for the Classification of Human Sperm Head Morphology. Advanced Intelligent Systems 4(10), (2022). https://doi.org/10.1002/aisy.202200111
11. Vermeulen, A. F.: Practical Data Science. 2nd edn. Apress (2018). https://doi.org/10.1007/978-1-4842-3054-1
12. Wen, L., Li, X., Gao, L.: A transfer convolutional neural network for fault diagnosis based on ResNet-50. Neural Computing and Applications **32**(10), 6111–6124 (2020). https://doi.org/10.1007/s00521-019-04097-w
13. Yüzkat, M., Ilhan, H. O., Aydin, N.: Multi-model CNN fusion for sperm morphology analysis. Computers in Biology and Medicine **137**, 104790 (2021). https://doi.org/10.1016/j.compbiomed.2021.104790
14. Zeng, G.: On the confusion matrix in credit scoring and its analytical properties. Communications in Statistics - Theory and Methods **49**(9), 2080–2093 (2020). https://doi.org/10.1080/03610926.2019.1568485