

UKRAINIAN CATHOLIC UNIVERSITY

BACHELOR THESIS

End2end image analysis of single-cell gel electrophoresis

Author:
Mariya HIRNA

Supervisor:
Igor KRASHENYI

*A thesis submitted in fulfillment of the requirements
for the degree of Bachelor of Science*

in the

Department of Computer Sciences
Faculty of Applied Sciences



APPLIED
SCIENCES
FACULTY ●

Lviv 2020

Declaration of Authorship

I, Mariya HIRNA, declare that this thesis titled, "End2end image analysis of single-cell gel electrophoresis" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all the main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

“Mais les yeux sont aveugles. Il faut chercher avec le cœur.”

Antoine de Saint-Exupéry

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

End2end image analysis of single-cell gel electrophoresis

by Mariya HIRNA

Abstract

Single-cell gel electrophoresis is the standard test used by biomedical researchers to analyze damage to the cell. Currently, this test is only done using standard image processing techniques, that skews the outputs, requires manual work and/or human supervision. Other problems with current solutions include poor usability, lack of flexibility, and high price for commercial applications. In this work, we create a deep learning-based end2end pipeline, that receives images from the test as an input, and produces damage metrics as an output. We have trained UNet with SE-ResNet50 encoder on the custom-created synthetic dataset, which achieves the dice coefficient of 76.8. We hope that the results of this work will become the base of the easy-to-use open-source application available for any researcher.

Acknowledgements

I want to express my appreciation to my supervisor - Igor Krashenyi, who guided me throughout this work, shared his experience and was always ready to help. I want to thank Olena Domanska and Lyubomyr Senyuk, who helped with the idea for the research, and their support. I want to thank Danylo Kolinko for his contributions and dedication.

Finally, I want to thank my family for their continuous love and support.

Contents

Declaration of Authorship	i
Abstract	iii
Acknowledgements	iv
1 Introduction	1
1.1 Goals	1
1.2 Thesis structure	1
2 Biomedical background	3
2.1 Single cell gel electrophoresis	3
2.2 Image Analysis	3
Distance of DNA migration	4
Damage classification	4
Tail moment	4
2.3 Usage	5
3 Technical background	7
3.1 Artificial neural networks	7
3.2 Convolutional neural networks	8
Convolutional layer	8
Pooling layer	8
Image classification	9
3.3 Semantic Segmentation	9
Segmentation using Deep Learning	9
Upsampling	10
Encoder-Decoder Architectures	11
4 Related work	13
4.1 Comet assay image analysis	13
Comet classification	13
Metrics calculation	13
4.2 Cell segmentation	15
5 Materials and Methods	17
5.1 Dataset creation	17
5.1.1 Data annotation	17
5.1.2 Synthetic dataset generation	17
5.1.3 Data Augmentation	18
5.2 Semantic Segmentation	19
5.2.1 Architectures	19
UNet	19
Attention R2UNet	19

SAUNet	20
5.2.2 Losses	21
5.3 Further processing	21
6 Results	23
6.1 Experiments	23
Experiment results	23
7 Summary	25
7.1 Future work	25
Data gathering	25
Open-source application	25
Bibliography	26

List of Figures

2.1	Comet-like cells after SCGE.	3
2.2	Five classes of comet damage by Collins	4
2.3	Damage metrics calculation	5
3.1	Artificial Neural Network.	7
3.2	Filters in Convolutional Neural Networks.	8
3.3	Max pooling	9
3.4	BBBC039 sample with instance and semantic maps.	9
3.5	Max unpooling	10
3.6	Bilinear interpolation	10
3.7	3x3 transpose convolution, stride 2, pad 1	11
3.8	UNet architecture	11
4.1	Comet segmentation pipeline.	14
4.2	Cellprofiler segmentation predictions.	14
5.1	Mask with synthetic overlaps.	18
5.2	Synthetic data with cells.	18
5.3	Augmented images	19
5.4	Residual Attention UNet	20
5.5	Shape-Attentive UNet	20
5.6	Edge Enhancement Loss	21
5.7	Watershed processing of the mask	22
6.1	Images with low results on predictions	24

List of Tables

6.1 Experiments with original dataset	23
6.2 Experiments	24

List of Abbreviations

ANN	Artificial-Neural Network
CNN	Convolutional-Neural Network
DNA	DeoxyriboNucleic Acid
SCGE	Single-Cell Gel Electrophoresis
SAUNet	Shape-Attentive UNet
CPH	Center Position of the Head
CMT	Center of Mass in Tail
UV	UltraViolet
ReLU	Rectified Linear Unit
FCN	Fully Convolutional Network
CV	Computer Vision
CVAT	Computer Vision Annotation Tool

Dedicated to my father

Chapter 1

Introduction

Single-cell gel electrophoresis is one of the most usable tests for cell damage assessment in the world. Though being very popular, it is not finally standardized, and is used by researchers according to their needs. The final part of the test is image analysis, in which segmentation is performed on the image of individual cells, and metrics of damage are calculated.

Medical image segmentation has improved during recent years due to usage neural networks and creation of new datasets. However, for single-cell gel electrophoresis there are no implementations involving deep learning, as well as no dataset for image segmentation. Image analysis is done using standard computer vision methods, that sometimes can not handle the complexities that occur in dataset, like overlaps, debris, different image characteristics. Open software, that produces resulting metrics is mostly old, poorly supported / not usable, not accurate, very restricted in usage, and requiring manual supervision. Commercial programs can be very expensive, starting from \$3000, so they couldn't be tested for this work.

1.1 Goals

In this work, we want to create and test a single-cell gel electrophoresis dataset for segmentation using the latest developments in medical image segmentation. We also want to process the resulting masks to create an end2end model for damage metrics calculation, which will become the basis of the image analysis application. We are curious to discover which results we can achieve using 125 samples hand-labeled dataset, annotated by not professionals.

1.2 Thesis structure

Chapter 2. Biomedical background

This chapter contains information on the process and usage of single-cell gel electrophoresis and approaches to image analysis of the microscopic slides.

Chapter 3. Technical background

In this chapter, we provide background on neural networks, convolutions, and semantic segmentation.

Chapter 4. Related work

This chapter includes the related work on comet assay image analysis, as well as recent developments in cell segmentation.

Chapter 5. Materials and Methods

Here we explain the dataset annotation, the process of creating the synthetic dataset, data preprocessing, and augmentation. We describe the models used for the experiments and metrics used for comet evaluation.

Chapter 6. Experiments

In this chapter, we describe the experiments conducted on the dataset and report the scores.

Chapter 7. Summary Here we summarize the work done, achieved results, and plans for future work.

Chapter 2

Biomedical background

2.1 Single cell gel electrophoresis

Single-cell gel electrophoresis is the most common technique for DNA damage measurement using visual damage of eukaryotic cells. SCGE is used for assessment of DNA strands on individual cell level requiring low-cost, short time for the test, and a small number of cells for sample per test. Single-cell gel electrophoresis has a high sensitivity that captures low-level DNA damage and is flexible, as it allows us to capture various types of cell damage.

The test is conducted the following way - suspension of cells is embedded in an agarose gel and put on the microscopic slides. The cells undergo lysis, the process of breaking down the membrane of the cell to purify and liberate the DNA. To convert some types of damages into DNA strands damage, and make them visible during the process, cells are put into alkali-labile sites during the alkaline unwinding stage. The next important step is electrophoresis, in which negatively charged parts of the DNA start moving towards the anode, migrating away from the nucleus and spreading to form a comet-like structure. These comet-like looking cells resulted in single-cell gel electrophoresis being referred to as comet assay.

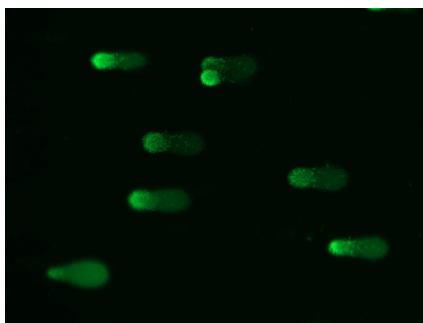


FIGURE 2.1: Comet-like cells after SCGE.

After the electrophoresis, alkali neutralization is performed. Succeeding step is to fix the comets and stain them with fluorescent or silver nitrate substance. Latter will result in noisy black comets, while fluorescent - light green ones. The final step is to capture images of the microscopic slides and send them for image analysis to calculate metrics describing each isolated comet.

2.2 Image Analysis

Though the methodology developed by Ostling and Johanson in 1984, the protocol for standardized comet evaluation was only developed in 2006. The procedure can

differ drastically, depending on the needs of the researchers using it. After receiving an image as an input, all image analysis programs will output one of the three results.

Distance of DNA migration

This simple approach is suitable for only relatively low damage to the cell. It measures the distance of DNA migration from the body of the nuclear core. It would not be useful for cases when DNA damage is substantial, as with increasing damage DNA, the intensity increases, but not the length of the comet.

Damage classification

Collins suggested assessing comets by 5 classes (0 to 4), in which 0 class would indicate no damage, while 4 - maximum damage for DNA strands. This approach is currently used in many labs, though it can not provide a more detailed description by each comet. It is common to calculate the average of classes per image to estimate the damage.

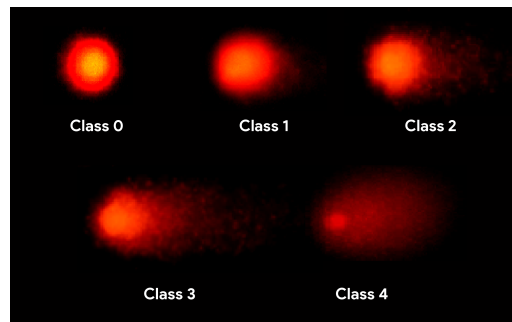


FIGURE 2.2: Five classes of comet damage by Collins

Tail moment

A more accurate and flexible approach is calculating metrics, describing the damage. The main idea is that DNA in a comet is proportional to the sum of intensity values of all the pixels representing the comet.

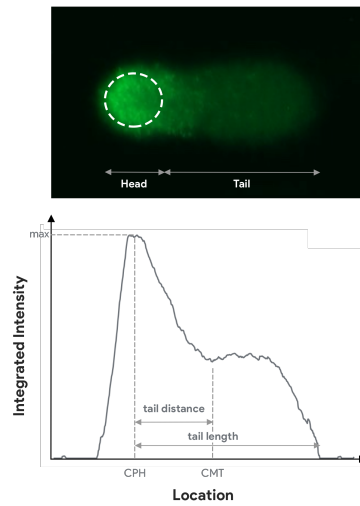


FIGURE 2.3: Damage metrics calculation

According to this idea, we can calculate DNA, and tail DNA ratio, as follows:

$$DNA = \sum_{x \in comet} I(x)$$

$$TDNA = \frac{1}{DNA} \sum_{x \in tail} I(x)$$

CPH is the center position of the head, and CMT is the tail center of mass. Tail length is the distance between the rightmost most point of the head, and the end of the comet. Tail distance is the distance between CPH and CMT. An example of these points on one of the comets from the dataset can be seen in figure 2.3. Having all the coordinates, we can calculate three more metrics describing the comet: extent moment, a tail moment of inertia, and Olive moment.

$$ExtentMoment = TailLength * TDNA$$

$$OliveMoment = TailDistance * TDNA$$

$$TailMomentInertia = \frac{1}{DNA} \sum_{x \in tail} I(x) * (CPH - x)^2$$

2.3 Usage

SCGE is used in biomonitoring, genotoxicology, ecological monitoring, DNA damage/repair assessment in response to DNA-damaging agents.

Hoeijmakers classified main DNA damaging agents into three categories Hoeijmakers, 2009:

1. environmental (UV light, ionizing radiation, genotoxic chemicals)
2. normal cellular metabolism

3. chemical agents that bind to DNA and cause spontaneous disintegration

Apart from working with agents, more specific cases were described by M, A, and S, 2018. They include testing of newly developed pharmaceuticals, research in diabetes, rheumatoid arthritis, Alzheimer's and Parkinson's disease, male infertility testing, detection of toxic environmental factors (radiation, heavy metals), evaluation of carcinogens, forecast of tumor radio and chemosensitivity. Due to the flexibility of the test, it is used in other research areas as a standard method to assess the damage.

Chapter 3

Technical background

Neural networks have become a standard in medical image segmentation due to the ability to learn complex patterns and features, state-of-the-art results, and the ability to generalize on different data. In this chapter, we will cover introductions to artificial neural networks, convolutional neural networks, and semantic segmentation.

3.1 Artificial neural networks

Artificial neural networks are an information processing paradigm, meant to mimic the human brain. ANNs consist of layers of nodes (neurons) that are connected in an acyclic graph. Each ANN would have an input layer, output layer, and hidden layers. The input layer merely contains input data, the output layer contains a prediction of a neural network, and hidden layers number and structure vary depending on the ANN architecture design. One of the most popular types of layers is a fully-connected layer, in which all neurons in two adjacent layers are pairwise connected. Depending on the specifics of the task (for example, classification, or regression), the output layer will have different dimensions.

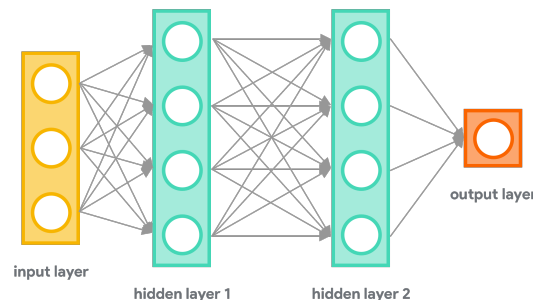


FIGURE 3.1: Artificial Neural Network.

Input in the neural network is a single vector that is processed by neurons. Each neuron is a mathematical function that maps its' input x into output y by combining weights with the input and passing it through the activation function. Weights can be interpreted as connections between neurons. They are learned and updated throughout the whole process of training the neural network. After completing the forward pass - passing the input through all the layers of the ANN, and saving the outputs, we will complete a backward pass - calculate the gradients and update the weights.

3.2 Convolutional neural networks

Convolutional neural networks are similar to ANNs in that they also consist of neurons. Neurons process the data by taking an input, computing the dot product, and sometimes passing it through the non-linear function. However, they are designed to process images and preserve the spacial information, by taking an input image as is - with its width, height, and depth dimensions, and not stretching the data into a single vector, as ANNs. Convolutional Neural Network for image classification will typically consist of the input layer, a convolutional layer, a layer with activation function, a pooling layer, and a fully-connected layer.

Convolutional layer.

Convolutional layer parameters consist of filters - matrices, whose elements are learned and updated during the training. We compute the dot product of the filter and the current part of an image, record the output, and slide filter to the next part.

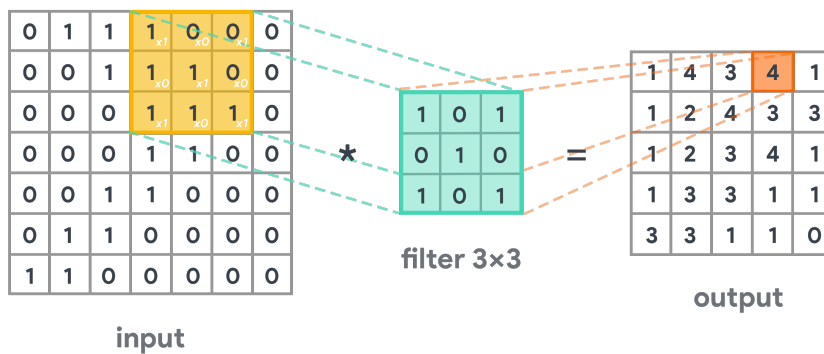


FIGURE 3.2: Filters in Convolutional Neural Networks.

To control the output of the convolutional layer, and to set the way filters to slide over the image, we can regulate the following hyperparameters: stride, number of filters, filter size, padding. Stride is the number of pixels we shift on an input image after each slide. Padding an input image may be useful to make filter fit over the input dimensions. The number of filters will impact the output depth. If W is an input image size (width, or height), F - is the filter size, P - is padding, and S is stride, then width and height of the output can be calculated by the formula:

$$(W - F + 2P) / S + 1$$

Pooling layer

The pooling layer is used for downsampling along the spatial dimensions. Its' function is to reduce the number of parameters, thus reducing the compute and sometimes to prevent overfitting. Different types of pooling would include max pooling, average pooling, and sum pooling. The former is the most common to use.

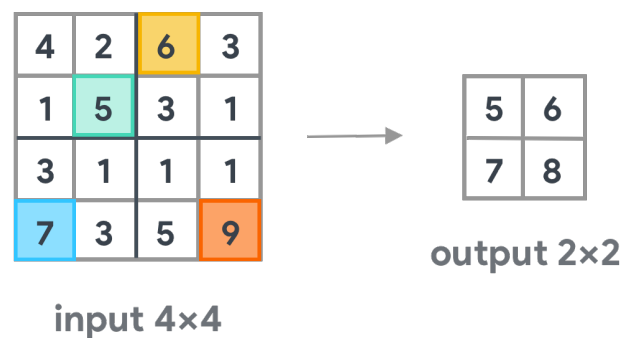


FIGURE 3.3: Max pooling

Image classification

For image classification problem, CNN would usually consist of input layer, combination of convolutional layer, ReLU, and pooling layer, and fully-connected layer in the end, with dimensions corresponding to number of classes.

3.3 Semantic Segmentation

Semantic segmentation is the task of linking each pixel of the image with the specific class. It could be thought of as pixel-level image classification because each pixel in an image is classified according to a category. In the case of cell segmentation with three classes: cell, nucleus, artifacts, each pixel would be labeled according to the category or labeled as background. The segmentation model would produce a mask as the output. However, overlapping nuclei would not be distinguishable in this setup, and would be look merged in the output mask. It is indistinguishable, as semantic segmentation is different from instance segmentation, and the model can not distinguish between each instance of the nucleus.

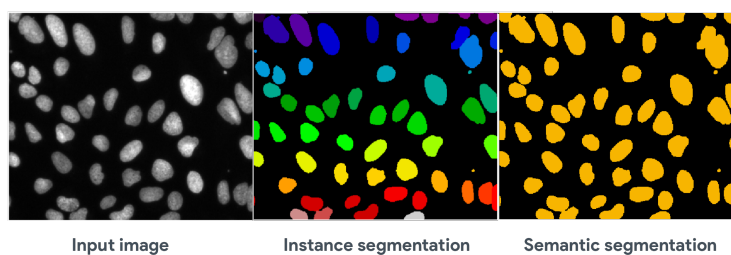


FIGURE 3.4: BBBC039 sample with instance and semantic maps.

Though having its limitations, semantic segmentation is widely used for scene understanding, autonomous driving, image editing tools, robotics, and biomedical image analysis.

Segmentation using Deep Learning

In 2012 Cirean et al., 2012 published Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images, in which they performed segmentation by classifying each pixel using a region around it and sliding through the image.

This approach was limited by the fact that regions (patches) were overlapping, resulting in redundancy and computational inefficiency. Additionally, they faced a trade-off between localization accuracy and use of context, as larger patches required more pooling, and thus lowered the localization accuracy. In Shelhamer, Long, and Darrell, 2016 introduced Fully Convolutional Networks for semantic segmentation. Their idea was to downsample, similarly to traditional CNN for image classification, remove the fully-connected layer in the end, upsample the feature bottleneck to image size, and predict a semantic map in the end. They used pre-trained classification networks like AlexNet, VGG, and GoogleNet, as powerful feature extractors, and modified them into FCNs for semantic segmentation. Compared to processing the image patch-by-patch, this approach significantly improved compute and accuracy. Further development in segmentation, was to change one-step upsampling operation to a series of upsamplings done in a few layers.

Upsampling

The simplest approach to upsampling is unpooling. In this case, we save the indices from max-pooling during downsampling, and during upsampling unpool into indices saved from the corresponding part of the network. Below is the example of max unpooling, also referred to as "Bed of nails."

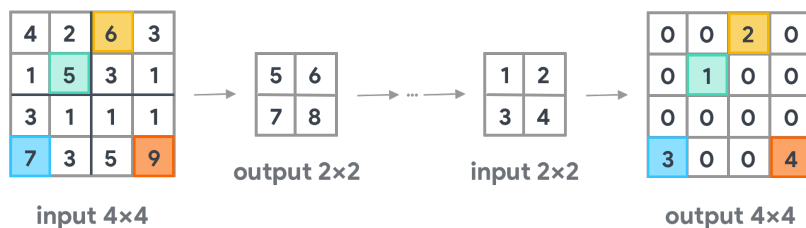


FIGURE 3.5: Max unpooling

In the case of interpolation, we will use a linear combination of neighboring pixels (two for bilinear, and four for bicubic), to calculate the value of the pixel. This type of upsampling produces smooth output.

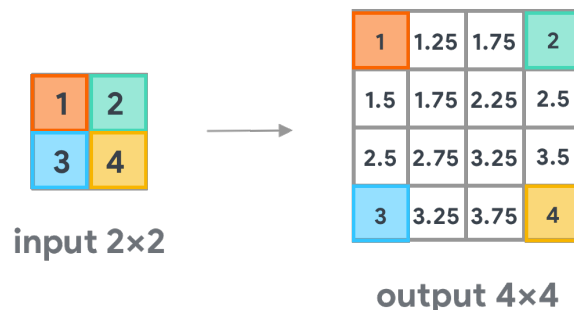


FIGURE 3.6: Bilinear interpolation

Previous upsampling methods did not include any learnable parameters, so another method for upsampling would be transpose convolution, also called deconvolution. The idea is to have filters, similarly, as in convolution, take each scalar value

in the input image, multiply it by the filter, and place it in the corresponding region of the output. Whenever outputs overlap, we sum the values.

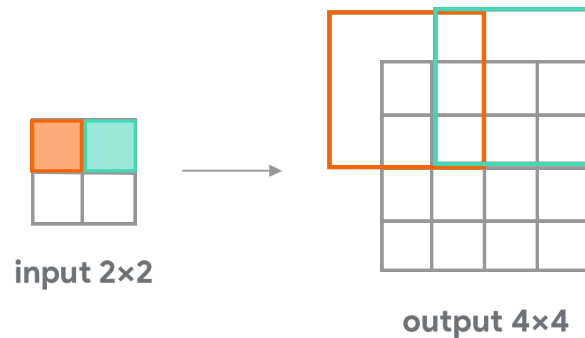


FIGURE 3.7: 3x3 transpose convolution, stride 2, pad 1

Encoder-Decoder Architectures

In semantic segmentation, a popular architecture with downsampling and upsampling is called encoder-decoder. The encoder will take an image and produce a high-dimensional features vector as an output. The decoder will take the feature bottleneck in, as an input, and produce a semantic map, as an input. This architecture was proposed by Badrinarayanan, Kendall, and Cipolla, 2015 in SegNet, where they changed upsampling in one step to decoder, and used unpooling for upsampling. Later, in the learning deconvolution network Noh, Hong, and Han, 2015, upsampling was done using transposed convolution or deconvolution.

The next important architecture for image segmentation is UNet.

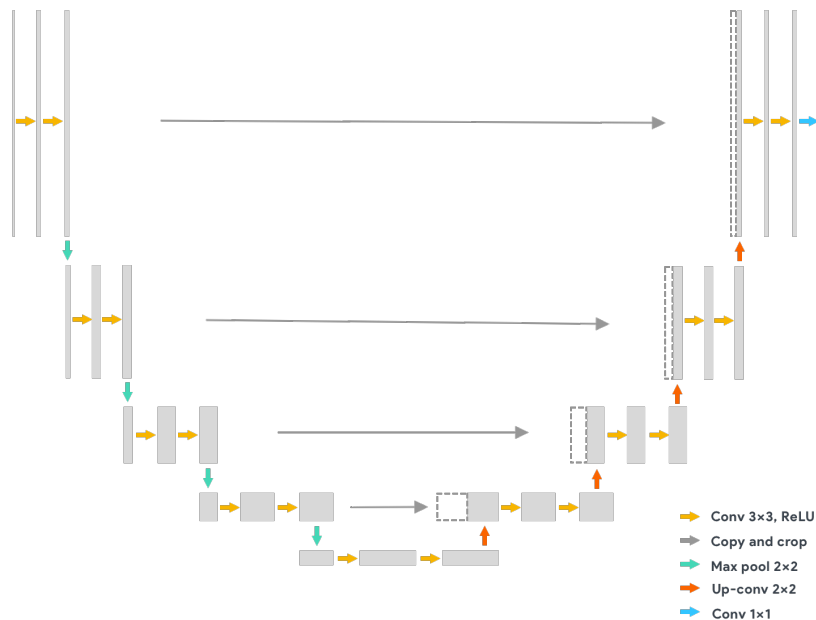


FIGURE 3.8: UNet architecture

UNet was specifically developed for biomedical image segmentation by Ronneberger, Fischer, and Brox, 2015 and is currently the base of most biomedical segmentation architectures. UNet consists of two paths: contracting and expanding, encoder, and decoder, which are connected by skip connections. Skip connections were introduced to handle small objects, and they used interpolation instead of transposed convolution to handle checkerboards artifacts and produce a smooth output.

Chapter 4

Related work

4.1 Comet assay image analysis

Comet classification

Despite being a very popular method, SCGE still has no open dataset for image analysis, allowing it to improve the current results. Some data available online was found on Afiahayati et al., 2020 Github repository. This is the only paper that incorporated neural networks for comet analysis. Their work is based on CNN usage for comet classification by Collins 2.2. They have achieved a classification accuracy of 70.5%, using pre-trained VGG16. They have tested their dataset on OpenComet Gyori et al., 2014 - the most popular free tool for comet analysis, and it was only accurate in 11.5%.

This approach can certainly automate hand labeling of the classes, like done in O. Yu. Harmatina, 2019, but is limiting the comet quantification to class. The input into the network, are comets, segmented by thresholding, so this approach is not flexible to work with images with different characteristics, overlapping comets, debris, and artifacts.

Metrics calculation

Works, that analysed images to compute tail moment, and other metrics 2.2, are Ganapathy et al., 2014, Ganapathy et al., 2015, Jones et al., 2008, Gyori et al., 2014, and Lee et al., 2018. They analyse comets using standard computer vision methods similarly to the following pipeline, which is typical for most tools for SCGE image analysis.

Though some of these solutions have an open-source implementation, most of them are not usable, as they are either not supported, have no documentation on installation, like in Lee et al., 2018, are not flexible for changes Jones et al., 2008, or sometimes require hand labeling the comet, or its' head, like *CaspLab*.

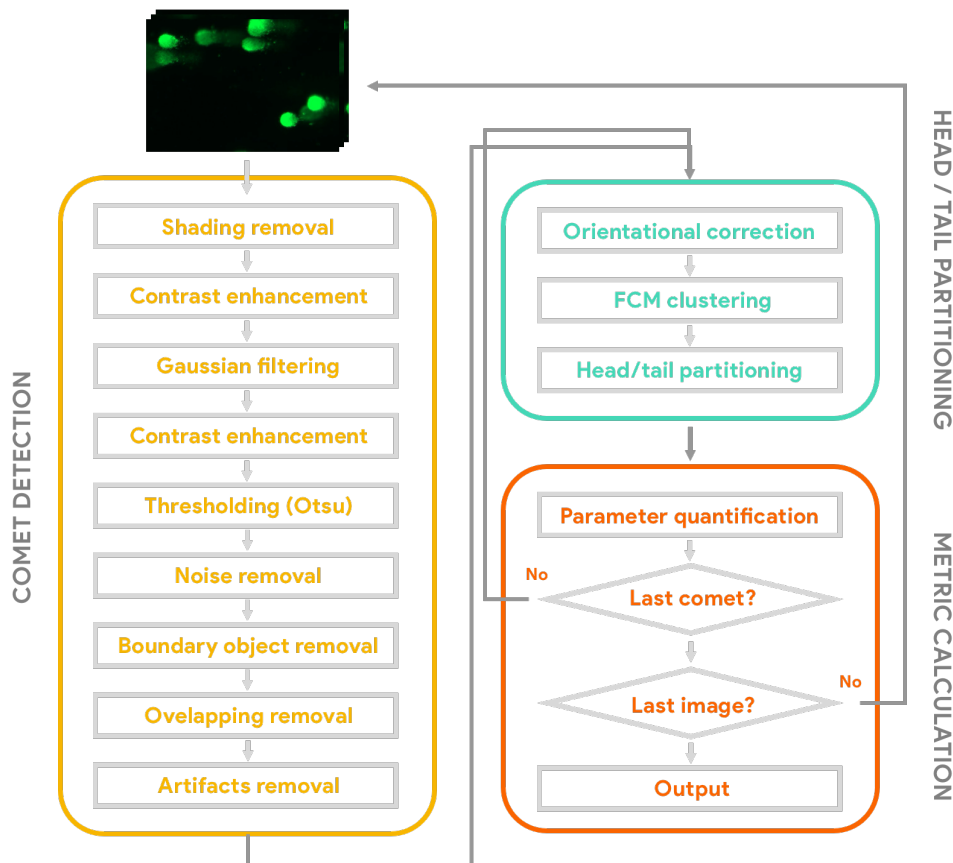


FIGURE 4.1: Comet segmentation pipeline.

Their approach to segmentation is based on Otsu's thresholding method, filtering by shape and simple CV techniques, like contrast enhancement or noise removal. However, depending on image size, specifics of data - overlapping, faded, blurred, not damaged comets, image characteristics, and quality, many parameters in this pipeline need to be fine-tuned to perform accurately. For example, when an image from our dataset was fed into cell profiler comet assay pipeline, it produced the following results:

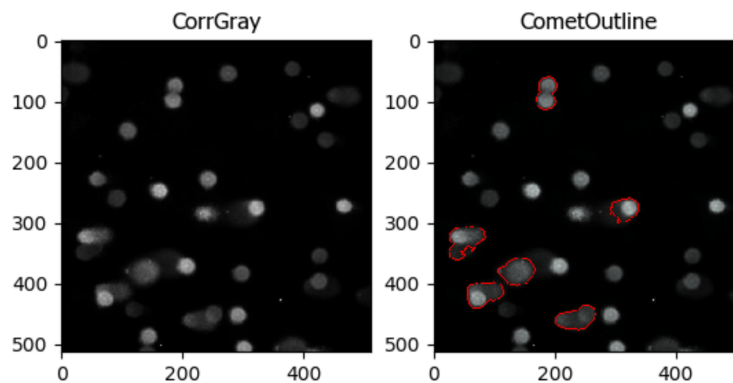


FIGURE 4.2: Cellprofiler segmentation predictions.

Overlapping comets are usually discarded, as they can not be handled by standard methods. Lee et al., 2018 in their work tackled this problem, and they correct the overlapping comets using watershed and distance transforms. After receiving the individual comets they validate them using the Fourier shape descriptor and the assumption, that all comets are elliptical.

As stated in Lee et al., 2018, "Although these platforms have various new aspects, the basic principle behind the analysis has remained unchanged: identification and characterization of individual comets. Due to the overlap of comets and debris, most of the existing analysis programs require laborious manual identification of comets from the fluorescent images."

Apart from filtering and handling overlaps, HiComet classifies DNA damage states into normal, necrosis, and apoptosis. Necrosis and apoptosis refer to the type of cell "death". Necrosis is unregulated cell death, as a result of external or internal stresses, while apoptosis is regulated cell death, triggered by physical, chemical, or biological factors. Their work was the first to solve the problem of overlapping and analyze current software. However, the predictions are still limited by user-defined head dimensions and need to be set for each comet, and their open-source app couldn't be tested due to the absence of documentation and errors.

4.2 Cell segmentation

As comet segmentation has not been done using deep learning, the closest problem in computer vision would be cell segmentation. Cell segmentation includes different approaches, depending on the input dataset. Some cells are processed similarly to the pipeline described before, using watershed and image processing. In case we need to segment more classes, like cell nuclei, membranes, overlapping instances, remove artifacts and debris, this simple approach becomes insufficient. In this case, we use deep learning, that can recognize complex patterns, and generalize on different data samples. Most deep learning approaches to segmentation are usually based on UNet or MaskRCNN model family. The former ones are used in Zhu et al., 2020, Sun et al., 2020, or Ibtehaz and Rahman, 2019, and latter was used in Johnson, 2018.

The high focus in recent papers in biomedical image segmentation has been on learning shape features. This could be achieved by creating a special loss function like in Chen et al., 2019, where they create a loss function that takes into account both length of the contour as well as the area of a mask. In work by Zhu et al., 2020 it was proposed to incorporate edge information twice, by adding outer and inner border loss to the loss function. The idea is to preserve shape information and use it for a more accurate prediction of boundaries. Their dataset included multiple cell shapes, from round to star-shaped, and they have boosted Residual Attention UNet performance with the edge-enhancement loss idea. Another approach is to create a separate shape stream, as Sun et al., 2020 implemented in shape-attentive UNet. They propose not only to use the separate stream for shape features, that can be built-in a model, but also to use the outputs for shape stream for interpretability.

As semantic segmentation models are unable to distinguish between instances, touching cells can become a problem. To solve the problem a separate class can be added, that would represent a cell border - Chen et al., 2016, cell centroids - Zhou et al., 2019, or somehow label the overlap. Another problem in biomedical segmentation is the lack of data. To solve this problem heavy augmentation and synthetic

dataset generation could be applied. For example winners of DSBowl 2018 used similar approaches for cell segmentation *Data Science Bowl 2018 1st place*. As part of the augmentation they copying nuclei on images to create overlapping instances and help the model to better learn borders between them.

Chapter 5

Materials and Methods

When comets are not properly diluted - described in (Braafladt, Reipa, and Atha, 2016), the same problem of touching cells will occur. However, due to specifics of the data, cells are not touching, but sometimes fully overlapping. Our dataset included complex images with overlaps, so we will also focus our work to distinguish between the instances.

5.1 Dataset creation

The images were gathered from different sources online, like Github repositories, and open-source apps with samples provided. They include , Jones et al., 2008, and Lee et al., 2018. We have contacted eleven researchers, that published comet assay related works to help with the data gathering and annotation, and received a few more images to expand our dataset. Gathered data included many samples with not distilled cells, so they couldn't be properly analyzed during image analysis. Other images included text or elements that needed to be photoshopped, cropped, or otherwise edited. The final dataset consists of 125 images of comets after the fluorescent DNA stain. The images contain a total of 959 comets, 16.5% of which are overlapping.

5.1.1 Data annotation

Data samples were annotated using an open-source tool CVAT Sekachev et al., 2020. Each pixel was hand-labeled, as one of the following categories: background, head, comet body, or overlap. Having an instance of each comet, we have generated a center of the mass class.

5.1.2 Synthetic dataset generation

To create a full pipeline, and extract instances of comets, we need to process the output of the semantic segmentation model. However, due to the lack of data and class imbalance, the model couldn't learn proper boundaries between overlapping / nearby comets, thus skewing the output metrics. The resulting dataset doesn't include many overlaps (126), so we needed to create a synthetic dataset with overlapping comets and heavier augmentation. Not overlapping comets are situated on the black background, so it allowed us to generate realistic data and complicate it as far as we wanted.

All comets, that were not already overlapping, were clustered by head/tail ratio, intensity, and size into four groups. For each isolated comet with 30% probability, we picked a random comet from the same cluster, scaled it to have similar dimensions, placed it randomly near the main comet, with a high probability to overlap.

To automatically create the overlap on the mask, we have the following approach. Whenever heads were intersecting, we have put a stretched segment on the diagonal of the rectangle of the intersection. In the other case, when tails were intersecting, we have created a border mimicking dominant comet shape.

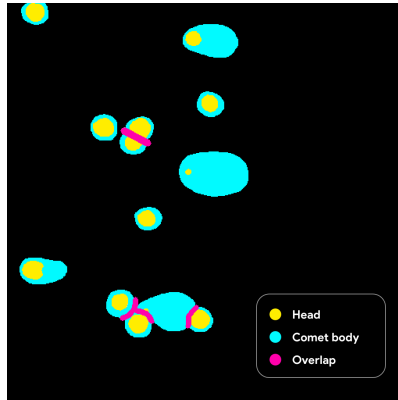


FIGURE 5.1: Mask with synthetic overlaps.

We have generated a synthetic dataset, and cleaned it from 4000 images to 1000 images, to include most realistic ones. To add more noise to the image, and increase the complexity, we have randomly placed nuclei of U2OS cells. We used image set Caicedo, 2018, available from the Broad Bioimage Benchmark Collection. The nuclei were augmented (randomly rotated, resized, blurred).

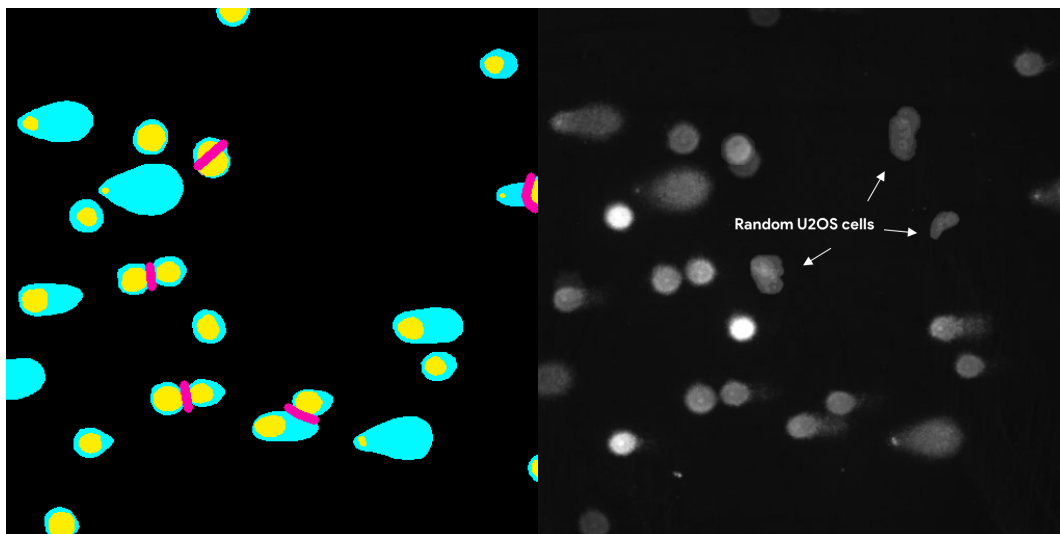


FIGURE 5.2: Synthetic data with cells.

5.1.3 Data Augmentation

We have created 9 hierarchical stages of data augmentation, from the lightest version to the heaviest one. For augmentation, we have used the augmentation library albumentations. Following is the list of the elements used in augmentations:

- Random size crop, flip, rotation
- Mask dropout - zeroing out random instances

- Random brightness /contrast
- Gaussian noise
- CLAHE
- Emboss, sharpen
- Affine augmentation
- Random sun flare, shaped like a circle
- Gray image to randomly colored
- Blur, median blur, and motion blur
- Addition of negative samples from other U2OS cells Bray et al., 2017

The last stage of augmentation was very similar to the one used in *Data Science Bowl 2018 1st place* for cell segmentation, and resulting images would look like shown on 6.1.



FIGURE 5.3: Augmented images

5.2 Semantic Segmentation

5.2.1 Architectures

In this section we will briefly describe the architectures used in experiments.

UNet

We have used UNet as a base architecture in our work. We have conducted experiments with multiple variations, like pre-trained VGG16 encoder, pre-trained ResNet34 encoder, and pre-trained SE-ResNet-50 encoder. UNet architecture was described earlier. 3.3.

Attention R2UNet

Attention R2UNet is a modification of UNet, built with two modifications. The first one is the addition of residual blocks, which were introduced by He et al., 2016, and the second one is the attention mechanism, which was introduced by Oktay et al., 2018. Both modifications were proposed as separate networks and later were

merged into one. All of the variations are highly popular for medical imaging, including pancreas segmentation, cell segmentation, and lesion boundary segmentation.

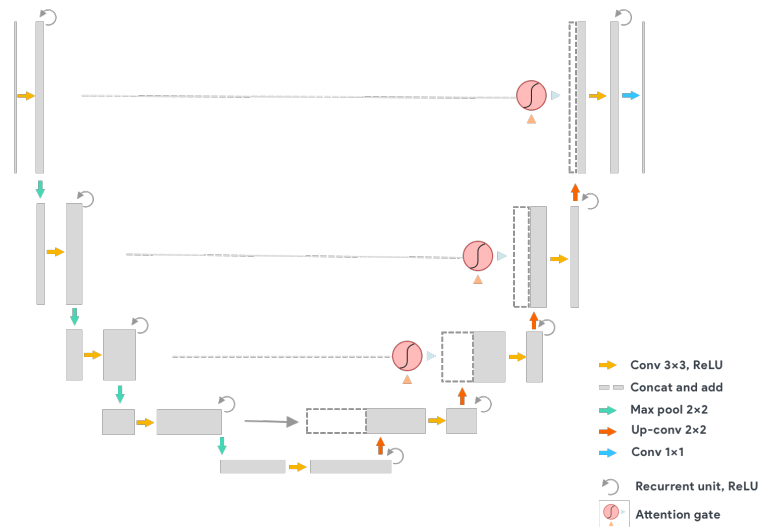


FIGURE 5.4: Residual Attention UNet

SAUNet

Shape-Attentive UNet by Sun et al., 2020 is another UNet modification, that consists of two streams - texture stream and shape stream. The modifications in texture stream include dense blocks and dual attention decoder blocks. Shape stream produces edge predictions that are incorporated into the loss function and can be descriptive for model interpretability.

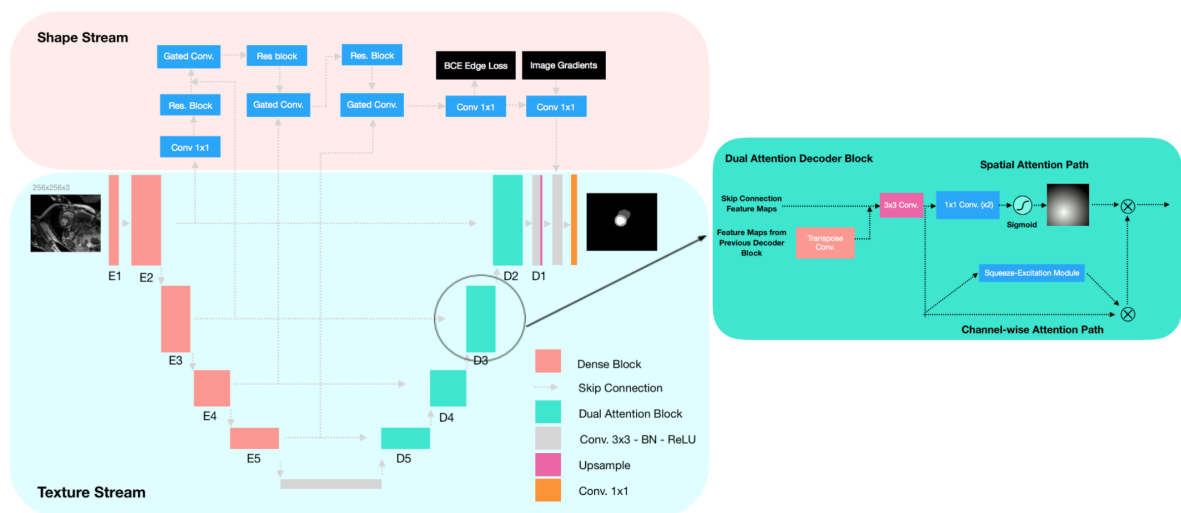


FIGURE 5.5: Shape-Attentive UNet

5.2.2 Losses

For training we have used standard losses for segmentation, like cross-entropy loss, dice loss, and a weighted combination of two. Dice loss is calculated as follows:

$$DL(p, \hat{p}) = 1 - \frac{2p\hat{p} + 1}{p + \hat{p}}$$

Also, we have used Edge Enhancement loss, that was proposed in Zhu et al., 2020, and is calculated, as described on the figure below.

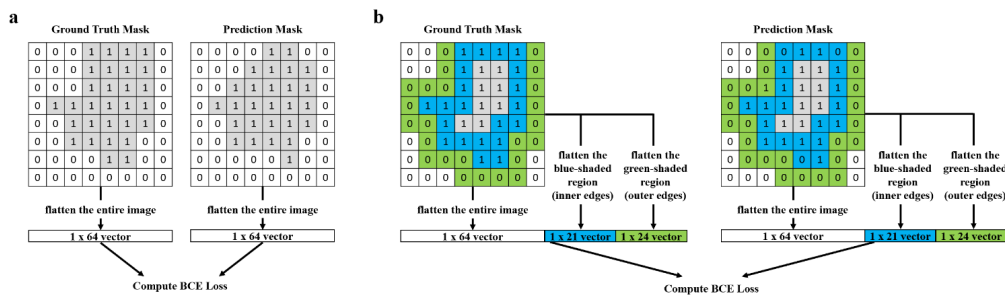


FIGURE 5.6: Edge Enhancement Loss

For SAUNet we have used dual loss, which is calculated as weighted sum of binary cross-entropy loss on edges, cross-entropy of segmentation prediction, and dice of segmentation predictions.

5.3 Further processing

To process the output semantic map into instances, we have used a watershed algorithm. It is a transformation on grayscale images in which pixel intensity is modeled as height on the topological map. We first take the output mask and threshold it to include heads and tails. Then we calculate the local gradients, which will have high pixel values along the edges. If we feed the gradients directly into the watershed, we can have over-segmentation, so we use markers to overcome this problem. Markers are components, connected to the image, that are of two types: internal marker, which denotes the object itself, and external marker, which denotes the boundary. Finally, we apply watershed to receive separate comets. The outputs of the predicted mask that is split into instances are shown below.

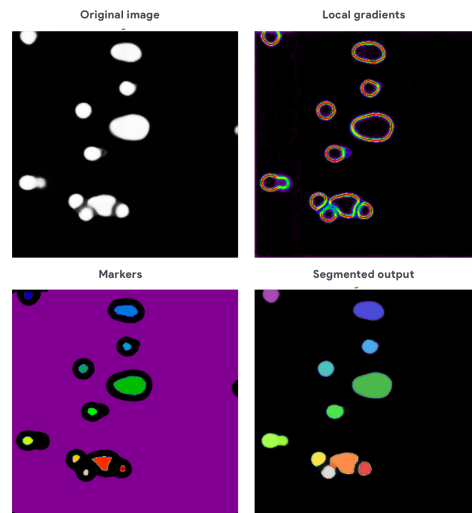


FIGURE 5.7: Watershed processing of the mask

Having segmented the instances, we calculate the metrics described above in [2.2](#) for final outputs.

Chapter 6

Results

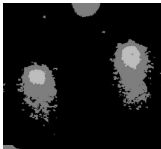
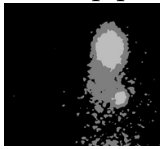
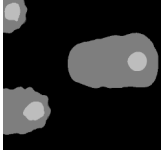
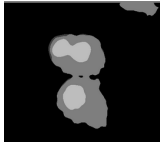
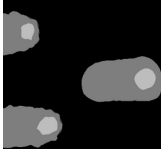
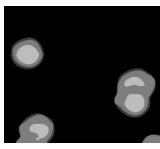
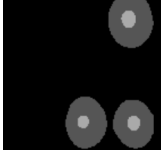

6.1 Experiments

Most experiments were done using Google Colaboratory, and we used neptune.ai for experiments storage.

Experiment results

Firstly, we were doing experiments using original dataset with 125 images described in 5.1, and the results are as follows.

TABLE 6.1: Experiments with original dataset

Model	Dice	Prediction	Overlap prediction
UNet with VGG16 encoder	35.1		
UNet with ResNet34 encoder	54.9		
R2Attention UNet + EE	53.2		
Shape-Attentive UNet	55.7		

Note: Dice is calculated for head and tail classes.

As one can notice, overlapping instances were not handled well by any of the models, even though separate comets features were learned well. To overcome this problem, we created a new class for comets overlapping, as described above, as well as generated a synthetic dataset that will include artificial overlapping. A chunk of our dataset included images that could be segmented using simple computer vision techniques. However, at least half of the images included either heavy overlaps, or debris, or some other type of artifacts, that needed to be removed. Some of the examples are below:

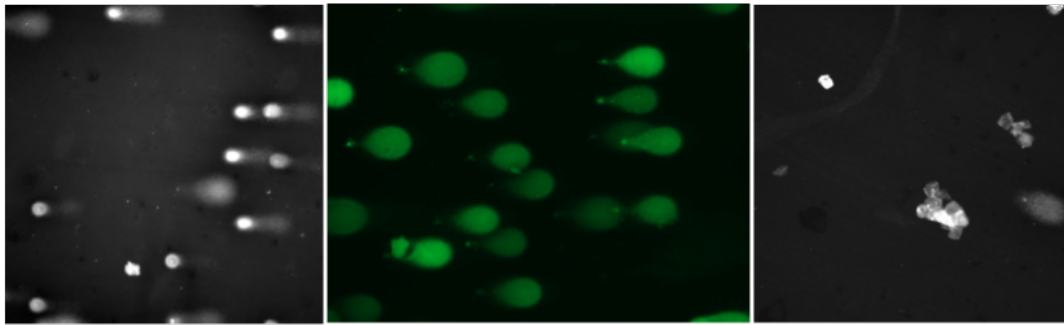
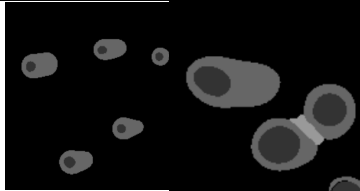

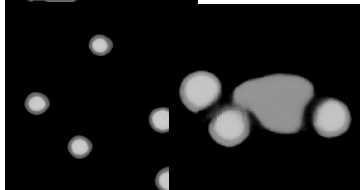
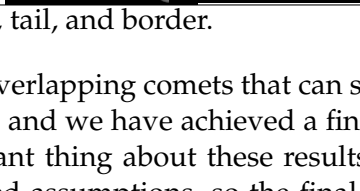
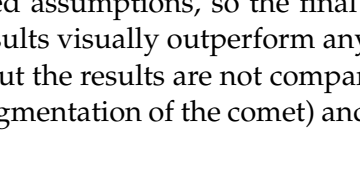



FIGURE 6.1: Images with low results on predictions

To train the model to learn how to separate overlaps, and to give predictions on images as above, we have created a synthetic dataset 5.1.2, cleaned it, applied heavy augmentation with other types of cells. After training on the new data we have received the following results.

TABLE 6.2: Experiments

Model	Dice	Prediction	Prediction overlap
UNet with VGG16 encoder	60.6		
Shape-Attentive UNet	59.1		
UNet with SE-ResNet50 encoder	76.8		

Note: Dice is calculated for all classes, head, tail, and border.

In final results, we are able to separate overlapping comets that can still produce separate metrics for the outcome of the test, and we have achieved a final dice metric on a validation set of 76.8. The important thing about these results is that the dataset included mistakes and comet-related assumptions, so the final accuracy is still limited by the dataset quality. These results visually outperform any segmentation available in open-source tools online, but the results are not comparable due to different approaches to segmentation (no segmentation of the comet) and because of no ground truth.

Chapter 7

Summary

In this work, we have created an end2end single cell gel electrophoresis pipeline. It consists of the following parts: image segmentation, further mask preprocessing, and metrics calculations. We have created the first work that uses semantic segmentation for comet analysis, as well as we were the first to create a corresponding dataset. After testing multiple solutions online, we have learned why some comet assay image analysis is still not automated and is usually fully-processed by humans or needs supervision. We hope that the problems learned and solved in this work will be further used to create an accurate tool that can be freely used by any researcher.

7.1 Future work

The next steps in our work will be as follows.

Data gathering

We have emailed multiple researchers all over the world, and plan to continue until we gather a larger dataset. The other help needed is supervision for data annotation, as we did all the annotations, and thus they are not too reliable.

Open-source application

We plan to continue working on this pipeline to create an open-source app for biomedical researchers that will be flexible to their requests and produce accurate results.

Bibliography

- Afiahayati, Afiahayati et al. (Jan. 2020). “Comet Assay Classification for Buccal Mucosa’s DNA Damage Measurement with Super Tiny Dataset Using Transfer Learning”. In: pp. 279–289. ISBN: 978-3-030-14131-8. DOI: [10.1007/978-3-030-14132-5_22](https://doi.org/10.1007/978-3-030-14132-5_22).
- Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla (Nov. 2015). “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP. DOI: [10.1109/TPAMI.2016.2644615](https://doi.org/10.1109/TPAMI.2016.2644615).
- Braafladt, Signe, Vytas Reipa, and Donald H. Atha (2016). “The Comet Assay: Automated Imaging Methods for Improved Analysis and Reproducibility”. In: *Scientific Reports* 6.1. DOI: [10.1038/srep32162](https://doi.org/10.1038/srep32162).
- Bray, Mark-Anthony et al. (Jan. 2017). “A dataset of images and morphological profiles of 30,000 small-molecule treatments using the Cell Painting assay”. In: *GigaScience* 6. DOI: [10.1093/gigascience/giw014](https://doi.org/10.1093/gigascience/giw014).
- Caicedo (2018). *BBBC039v1*.
- CaspLab. *CaspLab*. URL: <https://sourceforge.net/projects/casp/>.
- Chen, Hao et al. (Nov. 2016). “DCAN: Deep Contour-Aware Networks for Object Instance Segmentation from Histology Images”. In: *Medical Image Analysis* 36. DOI: [10.1016/j.media.2016.11.004](https://doi.org/10.1016/j.media.2016.11.004).
- Chen, Xu et al. (June 2019). “Learning Active Contour Models for Medical Image Segmentation”. In: pp. 11624–11632. DOI: [10.1109/CVPR.2019.01190](https://doi.org/10.1109/CVPR.2019.01190).
- Cirean, Dan et al. (Jan. 2012). “Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images”. In: *Proceedings of Neural Information Processing Systems* 25.
- Ganapathy, Sreelatha et al. (Dec. 2014). “Automatic detection of comets in Silver stained comet assay images for DNA damage analysis”. In: *2014 IEEE International Conference on Signal Processing, Communications and Computing, ICSPCC 2014*, pp. 533–538. DOI: [10.1109/ICSPCC.2014.6986250](https://doi.org/10.1109/ICSPCC.2014.6986250).
- Ganapathy, Sreelatha et al. (Oct. 2015). “Quantification of DNA damage by the analysis of silver stained comet assay images”. In: *IRBM* 36, pp. 306–314. DOI: [10.1016/j.irbm.2015.09.006](https://doi.org/10.1016/j.irbm.2015.09.006).
- Gyori, Benjamin et al. (Jan. 2014). “OpenComet: An automated tool for comet assay image analysis”. In: *Redox biology* 2, pp. 457–65. DOI: [10.1016/j.redox.2013.12.020](https://doi.org/10.1016/j.redox.2013.12.020).
- He, Kaiming et al. (June 2016). “Deep Residual Learning for Image Recognition”. In: pp. 770–778. DOI: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- Hoeijmakers, Jan (Oct. 2009). “Hoeijmakers JDNA damage, aging, and cancer. N Engl J Med 361:1475-1485”. In: *The New England journal of medicine* 361, pp. 1475–85. DOI: [10.1056/NEJMra0804615](https://doi.org/10.1056/NEJMra0804615).
- Ibtehaz, Nabil and Mohammad Rahman (Sept. 2019). “MultiResUNet : Rethinking the U-Net architecture for multimodal biomedical image segmentation”. In: *Neural Networks* 121. DOI: [10.1016/j.neunet.2019.08.025](https://doi.org/10.1016/j.neunet.2019.08.025).

- Johnson, Jeremiah (May 2018). "Adapting Mask-RCNN for Automatic Nucleus Segmentation". In:
- Jones, Thouis et al. (Dec. 2008). "CellProfiler Analyst: Data exploration and analysis software for complex image-based screens". In: *BMC bioinformatics* 9, p. 482. DOI: [10.1186/1471-2105-9-482](https://doi.org/10.1186/1471-2105-9-482).
- Lee, Taehoon et al. (Feb. 2018). "HiComet: A high-throughput comet analysis tool for large-scale DNA damage assessment". In: *BMC Bioinformatics* 19. DOI: [10.1186/s12859-018-2015-7](https://doi.org/10.1186/s12859-018-2015-7).
- M, Fahim, Ahmed A, and Hussain S (Jan. 2018). "Single Cell Gel Electrophoresis and Its Applications in Different Fields". In: *Journal of Formulation Science Bioavailability* 01. DOI: [10.4172/2577-0543.1000115](https://doi.org/10.4172/2577-0543.1000115).
- Noh, Hyeonwoo, Seunghoon Hong, and Bohyung Han (May 2015). "Learning Deconvolution Network for Semantic Segmentation". In: *ArXiv*. DOI: [10.1109/ICCV.2015.178](https://doi.org/10.1109/ICCV.2015.178).
- O. Yu. Harmatina . Yu. Voznesenska, N. H. Hrushka . . Kondratska . H. Portnychenko (2019). "Sirtuins and DNA damage of neurons in experimental chronic cerebral hypoperfusion". In: *Patology*.
- Oktay, Ozan et al. (Apr. 2018). "Attention U-Net: Learning Where to Look for the Pancreas". In: oryanay.
- Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (Oct. 2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: vol. 9351, pp. 234–241. ISBN: 978-3-319-24573-7. DOI: [10.1007/978-3-319-24574-4_28](https://doi.org/10.1007/978-3-319-24574-4_28).
- Sekachev, Boris et al. (Apr. 2020). *opencv/cvat: Cuboids + bug fixes*. Version v1.0.0-beta.2. DOI: [10.5281/zenodo.3779176](https://doi.org/10.5281/zenodo.3779176). URL: <https://doi.org/10.5281/zenodo.3779176>.
- Shelhamer, Evan, Jonathon Long, and Trevor Darrell (May 2016). "Fully Convolutional Networks for Semantic Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, pp. 1–1. DOI: [10.1109/TPAMI.2016.2572683](https://doi.org/10.1109/TPAMI.2016.2572683).
- Sun, Jesse et al. (Jan. 2020). "SAUNet: Shape Attentive U-Net for Interpretable Medical Image Segmentation". In: topcoders. *Data Science Bowl 2018 1st place*. URL: <https://www.kaggle.com/c/data-science-bowl-2018/discussion/54741>.
- Zhou, Zibin et al. (Nov. 2019). "Joint Multi-frame Detection and Segmentation for Multi-cell Tracking". In: pp. 435–446. ISBN: 978-3-030-34109-1. DOI: [10.1007/978-3-030-34110-7_36](https://doi.org/10.1007/978-3-030-34110-7_36).
- Zhu, Nanyan et al. (Jan. 2020). "Segmentation with Residual Attention U-Net and an Edge-Enhancement Approach Preserves Cell Shape Features". In: