Unsupervised preprocessing to improve generalisation for medical image classification

Mathias Kirkerød, Rune Johan Borgli
Simula Research Laboratory, Norway
University of Oslo, Norway
mathiaki@ifi.uio.no, rune@simula.no

Vajira Thambawita, Steven Hicks, Michael Alexander Riegler, Pål Halvorsen
Simula Metropolitan Center for Digital Engineering, Norway
University of Oslo, Norway
{vajira, steven, michael, paalh}@simula.no

***Abstract*—Automated disease detection in videos and images from the gastrointestinal (GI) tract has received much attention in the last years. However, the quality of image data is often reduced due to overlays of text and positional data. In this paper, we present different methods of preprocessing such images and we describe our approach to GI disease classification for the Kvasir v2 dataset. We propose multiple approaches to inpaint problematic areas in the images to improve the anomaly classification, and we discuss the effect that such preprocessing does to the input data. In short, our experiments show that the proposed methods improve the Matthews correlation coefficient by approximately 7% in terms of better classification of GI anomalies.**

***Index Terms*—Machine learning, GAN, Autoencoder, Inpainting**

I. Introduction

In the field of computer vision, image-based disease detection has become a popular area of research. For example, algorithms based on deep neural networks have been used to automatically analyse the human digestive system for anomalies such as polyps, lesions and other common illnesses. This is important as the detection and removal of colon polyps is the main prevention method of colorectal cancer, which ranks within the top-three terminal cancer types for both men and woman [1]. Automatically detecting this disease goes a long way of aiding doctors to perform a more thorough analysis of their patients, and has the potential of saving lives. In addition to gastroenterology, we continue to see machine learning based classification systems appear in nearly every branch of medicine.

In recent years, deep learning based algorithms have become a popular method for solving these problems. Aided by the rapid advancement of computational power due to the efficiency of GPUs, deep learning has shown state-of-the-art performance across numerous fields, including medicine. However, deep neural networks are only as good as the data used to train them. Thus, data which contains artefacts such as text and overlays may negatively impact the performance of models trained on this data. This is particularly problematic in medicine, as the selection of datasets is often limited, and the datasets available may include artefacts from the software the doctors use to analyse the images/videos (e.g., overlays, text, and other information).

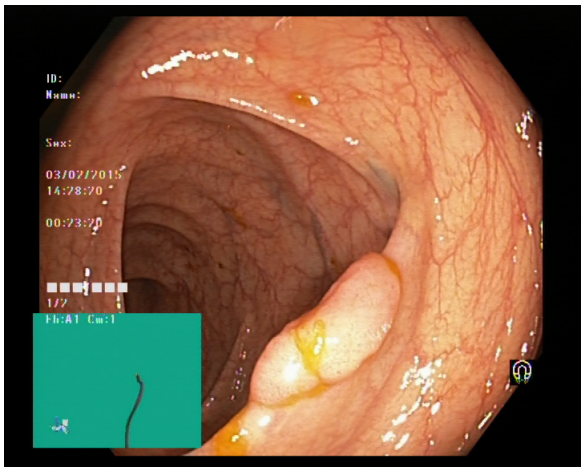


Fig. 1: Example image from the Kvasir dataset with included overlays and black borders.

In this work, we look at improving the quality of a publicly available endoscopy dataset called Kvasir [2], which contains several of the artefacts previously mentioned (example shown in Figure 1). We hope that this shows that there are more ways of improving the performance of a deep neural network than increasing its number of training samples. This work can be seen as an extension of our approach to this years MediaEval Medico task [3], where we presented a similar technique, albeit to a much lesser extent [4]. Additionally, a recent study using Kvasir for training deep learning based models showed that these artefacts directly impacted the classification performance of said models, showing that there is potential room for improvement [5].

The main contributions of this paper are (i) we present different methods for preprocessing data to be able to create better generalisable models, (ii) a detailed cross-dataset evaluation of the methods used and (iii) we report classification performance across different datasets.

II. Related Work

As mentioned in the introduction, medical image classification has been a heavily researched area. Research gathered by Lu and Weng [6] give current practices, problems, and prospects of image classification.

Our methods for inpainting bears a resemblance to context-encoder made by Pathak et al. [7] who introduce an encoder-decoder network in style close to our proposed generative adversarial network. However, a big difference is the use of a channel-wise fully connected layer in their model to share information around in the image space. This part was not necessary for us, given the homogeneity of the medical images coupled with the use of a non-random filter for inpainting.

Denton et al. [8] presented a model for inpainting close to the context-conditional adversarial network presented by Pathak et al. that is also trained on non-medical images, with random filter placement during training and evaluation. Their results showed that their generative adversarial network (GAN) model was capable of producing semantically meaningful inpaintings in a diverse set of images.

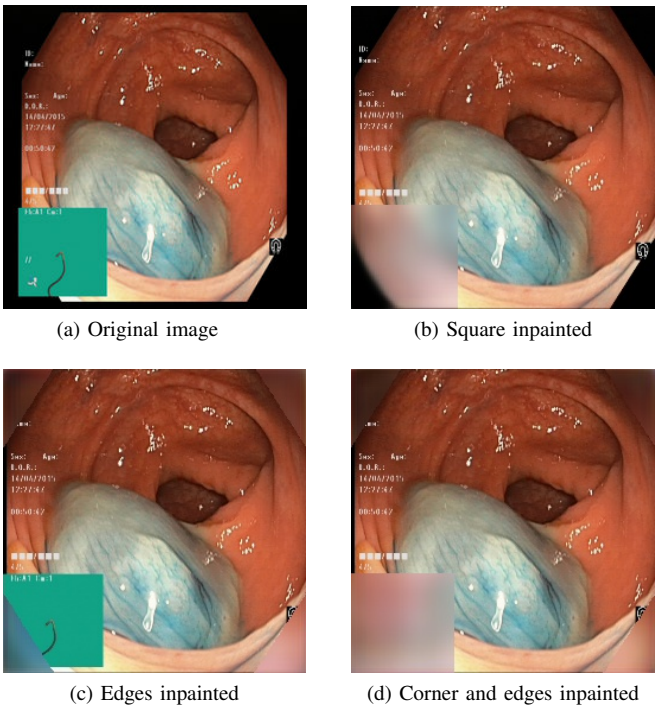
Previously, Hicks et al. [5] applied various preprocessing steps to Kvasir based on analysis conducted on common CNN architectures. Using heat maps and saliency maps, they discovered a common issue where artefacts such as text, black borders, and green navigation boxes were directly correlated to the misclassification of some images. In an attempt to correct this issue, they applied various preprocessing steps to the training data, namely cropping black borders and blacking out the green navigation box. Their results revealed improvement in all cases of data preprocessing, and in the best case, they achieved an increase of Matthews correlation coefficient (MCC) by approximately 3%.

In this paper, we aim to improve on this work by not simply removing borders and green navigation boxes. We also try to replace the artefacts using ideas from GAN inpainting to generate an automatically generated mask which attempts to replicate what would have been there if not for said overlay artefacts.

III. Approach

By using machine learning, we aim to classify medical images from the gastrointestinal (GI) tract correctly. With this approach, it is common to use a dataset for training and validation, with a separate set for testing. In practice the dataset we test on is never seen by the model before its evaluation. This is the main reason why we often struggle to get the same level of accuracy when evaluating our model if the data originates from different sources. In our case, the test data from the CVC dataset differs from the training data in both the image content and size. When this problem arises, it is practice to use domain-specific knowledge to help training, and if the amount of training data is small, methods like K-fold cross-validation [9] can also be used to improve the results.

For this paper, we focus on inpainting as a form of generalised preprocessing. We do this to remove dataset specific overlays for better classification on new datasets no matter the source of the dataset. Furthermore, we have also chosen to use the same Bayesian optimisation techniques as in the Borgli et al. paper [10] to find the optimal network for classification. With both hyperparameter optimisation and inpainting, our goal is to get the highest classification score on the CVC datasets.



(a) Original image (b) Square inpainted

(c) Edges inpainted (d) Corner and edges inpainted

Fig. 2: Here we have a sample of what we want to achieve. (a) Original from the Kvasir dataset. Here we also see extended edges that we can cut away without any machine learning. (b) Same image without edges and the green square. (c) Same image with new corners, (d) Same image with both new corners and new area for green square

A. Preprocessing

As discussed, the Kvasir dataset has some unwanted artefacts that are present in a good portion of the data. Some of the unwanted artefacts are Kvasir specific, and some are general artefacts when capturing images from the colon. First, the camera used in colonoscopies has an exceptionally wide lens. This setup takes good medical images but comes with the drawback that the images are not rectangular. Because of this, the camera needs to add black corners and borders to save the images. Another unwanted artefact that is Kvasir specific is an unwanted additional overlay added to the images. They are added post-image-capture by the medical staff, and they show essential information about the patient. As we can see from this, we have multiple areas in the images with pixels not originating from the patient, and subsequently contains no information relevant for classification.

A neural network will also often struggle with areas with really sparse information. Because of this, we believe that just replacing the green area with a similar black area will not yield the best result. However, we expect improvement if we instead

TABLE I: Details of all datasets used in the experiments.

BC: Black corner. **GS**: Green square. **BC+GS**: Black corner and Green square

Dataset labels	Size	Inpainted area	Generator network used
D-I	256x256 px	-	-
D-II	256x256 px	BC	Autoencoder
D-III	256x256 px	GS	Autoencoder
D-IV	256x256 px	BC+GS	Autoencoder
D-V	256x256 px	BC	GAN
D-VI	256x256 px	GS	GAN
D-VII	256x256 px	BC+GS	GAN
D-VIII	512x512 px	-	-
D-IX	512x512 px	BC	Autoencoder
D-X	512x512 px	GS	Autoencoder
D-XI	512x512 px	BC+GS	Autoencoder
D-XII	512x512 px	BC	GAN
D-XIII	512x512 px	GS	GAN
D-XIV	512x512 px	BC+GS	GAN

try to inpaint both the green corner and the black edges with data gathered from similar images. Furthermore, by removing areas that are specific for that dataset, we believe the model will be far better at generalising to other datasets within the same domain. In our case, the area we will be inpainting is the green area, since it is not present in the CVC datasets, and most other medical datasets are also without it.

With our two hypotheses, we have two different features that we believe will make the classification harder. We first aim to inpaint both areas separately to see how each of them affects classification. We also want to try to collectively remove both areas to see if a combined mask will yield a better or worse result.

With this in mind, we use two different methods for inpainting the desired areas. First, an autoencoder (AE) [11] as a lightweight way to generate new data, and second we use a GAN [12] as a more sophisticated generator. Both methods are unsupervised learning methods to generate new data within the distribution of the original dataset.

For our experiments, we scale our data to a constant resolution. We run four experiments with 256x256 pixels (px) resolution, and four experiments at 512x512 px. Our change in resolution is to compare the effect it has compared to our standard 256x256 px. With this configuration, we end up with 14 augmented datasets shown in table I.

B. Classification

Our research from the 2018 MediaEval workshop showed less desirable result compared to other projects that researched on the same dataset [13] [10]. Therefore one of our goals is to make our model more realistic by using a model that works better on the augmented Kvasir dataset. Using the Bayesian hyperparameter optimiser on our newly created datasets, we choose Densenet121 [14] as our default architecture for training our new datasets. We are also interested in the accuracy compared to a more general classification network. We ran model D-I - D-VII with the pretrained InceptionResNetV2 [15] network. We chose this network because of its high accuracy on the Keras websites [16], and thus we hypothesise that the model will be generally good without hyperparameter optimisation. In both cases, we remove the top layer and replace it with a global average pooling layer and a dense eight layer output to match the number of classes in the training dataset.

TABLE II: Details of experiments.

Test	Training datasets	Testing dataset	Network model
T1	D-I - D-VII	Kvasir V2	DenseNet121
T2	D-I - D-VII	CVC-12k	DenseNet121
T3	D-I - D-VII	CVC-356	DenseNet121
T4	D-I - D-VII	CVC-356	InceptionResnetV2
T5	D-VIII - D-XIV	CVC-356	DenseNet121

Our focus is the comparison between the generated datasets and the baseline; hence we do not change the hyperparameters after they are chosen. We believe this sets up a valid comparison since the only difference in score should come from the differences in the dataset and not the classification model. An overview of our experiments are shown in Table II, where Models T1 - T3 is a direct comparison on how well we have generalised our model, while Models T4 & T5 show how changing models will affect the results. Below, we give brief a description of the three datasets used.

a) The Kvasir V2 dataset [2]: The Kvasir V2 dataset consists of 8,000 images from the GI tract. Several of these images contain artefacts such as navigation boxes (green box as seen Figure 1), overlayed text, black borders, and black edges. With our first hypothesis in mind, we assume that the dataset with the inpainted rounding corners (D-II & D-IV) will do slightly better than the baseline (D-I). This is because the training and test data is from the same set, and subsequently our generalisation will not help. That leaves us with the only way to improve the result is to remove sparseness.

b) The CVC-356 dataset [17]: The CVC-356 dataset consists of 2,285 images from the lower GI tract. CVC-356 does not have images with green boxes. It does have images with black borders, and rounded black edges. As stated in our second hypothesis; the inpainting of the green square will presumably give the best result. This is because, as stated, the CVC-356 images has the same black rounded corners as Kvasir, but lacks the green squares.

c) The CVC 12k dataset [17]: The CVC-12k dataset consists of 11954 images from the lower GI tract, with a resolution of primarily 288x384 px. Given the similarity with the CVC-356 dataset, this will presumably follow our second hypothesis stating that the inpainting of the green square would give the best result. Given that the CVC-12k images has the same black rounded corners as Kvasir, but lacks the green squares.

IV. Preprocessing tools

The networks used for inpainting are based on the network presented in the Mediaeval conference [4]. Both networks are using on masking, where only the parts of the image corresponding to a mask was inpainted.

A. Autoencoder

The first approach we created and trained was a custom autoencoder [11] from scratch. Our autoencoder consists of an encoder-decoder network, with 2D convolutions as well as rectified linear units as activation functions, and a 25% dropout between the encoder and decoder. The network used is a modification of the network presented in [4]. The modifications are a smaller batch size and a more consistent filter size throughout the network. These modifications were made to make more credible results, and to get a lower error during training. The loss function was also modified to solely train on parts of the images that were modified. This lead to a larger and more accurate gradient descent, which also contributed to a better reconstruction.

B. Context conditional generative adversarial network

For the GAN approach, we create a similar structure to the autoencoder. We have a generator-discriminator network that serves much of the same functionality as the encoder-decoder network in the autoencoder. As with the autoencoder, we have the same size input as output, but we only decide to keep the parts we want to inpaint. The model we ended up with is closely inspired to the model made by Denton et al. in [18]. The main differences are the number of layers used, and the lack of a low-resolution image as an extra input.

V. Results

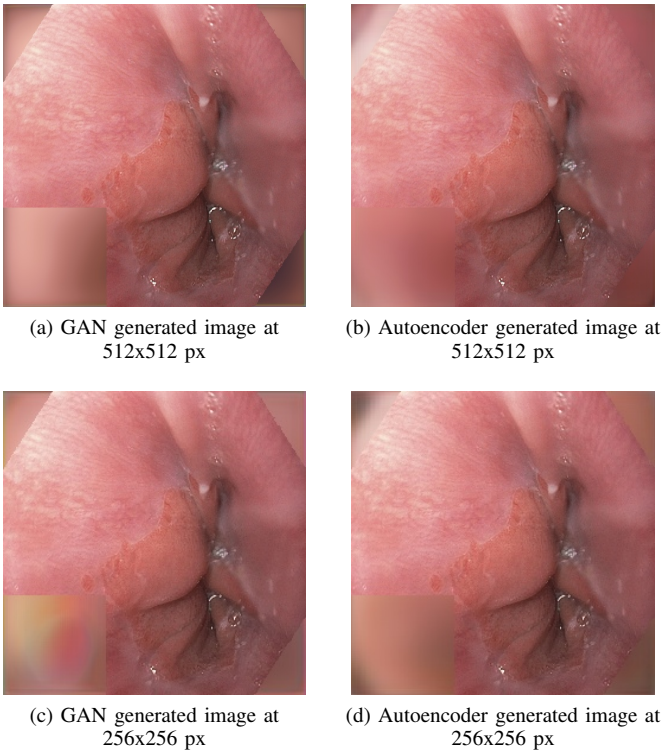
We divide our results into two sections, preprocessing and classification. In our preprocessing section, we discuss the appearance of the dataset, and how close the results are to the ground truth. In our classification section, we discuss the rate of generalisation and rate of success.

A. Preprocessing

Since there are no specific metrics associated with the training of Autoencoders and GANs, we used the mean square error of the ground truth as a metric of our progress. Figure 3 from the z-line shows how the two different models perform on the two different sizes. This is a typical case where both the GAN and the AE are fairly similar, except for more features added by the GAN. The features are most present in the smaller images, as the images are easier to train on, and subsequently easier to add complex local features too.

B. Classification

We evaluated our model on both the Kvasir and the CVC dataset as described in the classification section (III-B). When presenting our results, our main point of comparison is the MCC [19]. In addition to the MCC score, we use F1, precision and recall as metrics when presenting our results. In addition to the best MCC score, we present the average MCC score as an indicator of the general success of the method in question.



(a) GAN generated image at 512x512 px

(b) Autoencoder generated image at 512x512 px

(c) GAN generated image at 256x256 px

(d) Autoencoder generated image at 256x256 px

Fig. 3: Same image from the z-line with four different inpainting attempts. Each image is re-sized to fit in the figure.

Since our task was to improve classification and cross-dataset generalisability through inpainting, each table has its first row as the dataset without any inpainting, followed by the rest of the datasets. The first column is the maximum MCC score of the runs. Then we give the maximum F1 score followed by the maximum precision and recall. The last column gives us the average MCC of all four runs for each model.

First, we evaluated our results on the three datasets: Kvasir, CVC-356 and CVC-12k. Here our goal was to see the general improvement based only on inpainting and dataset. Then we evaluated the InceptionResNetV2 network on the CVC-356 dataset, and lastly, re-evaluated the CVC-356 network, at double image size.

a) Kvasir, Test T1: These are our results from training and evaluating on the Kvasir v2 dataset with the 5,600 image training set, 800 image validation set, and 1,600 image test set split. Table III shows the highest value for each of the six methods compared to the highest baseline.

As we can see in the results shown in Table III, we got the highest MCC score on the baseline dataset. Both the best and average scores were highest for the baseline, but the average was consistently high for all methods. As we recall, we predicted that we expected a higher MCC score for the Autoencoder inpainting the black corner and the GAN inpainting the black corner. The results do not show a clear

TABLE III: Test T1, Kvasir dataset on DenseNet121

Dataset	MCC	F1	Precision	Recall	Average MCC
D-I	0.9307	0.9394	0.9396	0.9394	0.9163
D-II	0.9150	0.9254	0.9303	0.9250	0.9053
D-III	0.9212	0.9310	0.9347	0.9306	0.9040
D-IV	0.9187	0.9287	0.9298	0.9288	0.9105
D-V	0.9208	0.9308	0.9316	0.9306	0.9108
D-VI	0.9096	0.9204	0.9226	0.9206	0.9055
D-VII	0.8960	0.9094	0.9174	0.9081	0.8926

TABLE IV: Test T2, CVC-12k dataset on DenseNet121

Dataset	MCC	F1	Precision	Recall	Average MCC
D-I	0.2897	0.5558	0.6968	0.6067	0.2723
D-II	0.3031	0.5413	0.7148	0.5927	0.2675
D-III	0.3197	0.6152	0.7050	0.6600	0.2649
D-IV	0.2956	0.4663	0.7632	0.5156	0.2733
D-V	0.2967	0.5451	0.7072	0.5965	0.2523
D-VI	0.2803	0.4548	0.7571	0.5038	0.2244
D-VII	0.2225	0.5740	0.6451	0.6236	0.1984

indication that the baseline was the best method, nor that there are any good ways to inpaint this dataset.

b) CVC-12k, Test T2: The T2 test case was trained on the Kvasir v2 dataset with the 5,600 image training set and the 800 image validation set, then evaluating on the CVC-12k dataset. Table IV shows the highest value for the six methods compared to the highest baseline, with four runs each.

As we can see in the results, shown in Table IV, we got the highest MCC score on the dataset with the inpainted green square made by the autoencoder. Also, the average score was consistently higher for the autoencoder datasets compared to the GAN datasets. The results give a small indication that inpainting the green area with an autoencoder might give a better result compared to the baseline.

c) CVC-356, Test T3: The T3 test case was, as test case T2, trained on the Kvasir v2 and evaluated on the CVC-356 dataset. The table V shows the highest value for each of the six methods compared to the highest baseline, with four runs each.

As we can observe in the results shown in Table V, we got the highest MCC score on the dataset with the inpainted green square made by the autoencoder and the GAN. We can also see a constant higher value for both datasets inpainting the green area. The highest value was from the dataset with both corner and square inpainting, but this is most likely just a lucky result, given the low average MCC. The results give a reasonable indication that inpainting the green area will give a better result compared to the baseline.

d) InceptionResNetV2, Test T4: These are our results from training on the Kvasir v2 dataset with the 5,600 image training set and the 800 image validation set, then evaluating on the CVC-365 dataset. The table VI shows the highest value for each of the six methods compared to the highest baseline, with four runs each. In this run we used the InceptionResNetV2 network to train our model.

TABLE V: Test T3, CVC-356 dataset on DenseNet121

Dataset	MCC	F1	Precision	Recall	Average MCC
D-I	0.7070	0.9137	0.9132	0.9164	0.5904
D-II	0.5153	0.7846	0.8153	0.8065	0.4861
D-III	0.7325	0.9402	0.9535	0.9348	0.6465
D-IV	0.6631	0.9264	0.9410	0.9194	0.5637
D-V	0.5714	0.8387	0.8487	0.8516	0.4557
D-VI	0.7150	0.9214	0.9206	0.9225	0.6334
D-VII	0.7466	0.9370	0.9391	0.9356	0.4576

TABLE VI: Test T4, CVC-356 dataset on InceptionResNetV2

Dataset	MCC	F1	Precision	Recall	Average MCC
D-I	0.4038	0.8851	0.9130	0.8678	0.2999
D-II	0.2221	0.7957	0.7958	0.7955	0.1227
D-III	0.0745	0.4489	0.5535	0.5131	0.0299
D-IV	0.3147	0.7793	0.7730	0.7916	0.1636
D-V	0.1802	0.5434	0.6201	0.5985	0.0446
D-VI	0.3276	0.8372	0.8429	0.8323	0.2234
D-VII	0.2738	0.6754	0.6938	0.7106	0.1417

As we can see from the results shown in Table VI, we got the highest MCC score on the baseline dataset. From our tests, it looked like the overall scores were much lower here compared to our DenseNet121 models, and in general, we got more unpredictable scores.

e) Double image size, Test T5: These are the results from training on the Kvasir v2 dataset with the 5600 image training set and the 800 image validation set, then evaluating on the CVC-365k dataset. The table VII shows the highest value for each of the six methods compared to the highest baseline, with four runs each. Here we have doubled the size of the images for the training and evaluation set to see how size affects the results.

On the CVC-356 dataset at 512x512 px resolution, we see a generally lower MCC score compared to the same dataset at 256x256 px. Our best average results came from the dataset with both inpainted corners and inpainted squares, but it looks like the more inpainting, the better. The results give a small indication that inpainting large areas with sparse information might give a better result compared to the baseline, at least compared to smaller areas.

Overall, we can observe through all experiments that inpainting can both improve and worsen the results. In general, inpainting works best when applied in dataset specific artefacts that are not present in the test set.

VI. Discussion

Our first hypothesis was that removal of the black edges and corners around the images would result in a better classification and better generalisation. Our results also show that training and testing on the same dataset gave approximately the same MCC score, with and without corners. In addition, we observed that the removal of areas within the images with no relevant information did not give any better results, given the same training and test distribution. This was not the case when the images were up-scaled above their original size, as we saw

TABLE VII: Test T5, CVC-356 dataset with double resolution

Dataset	MCC	F1	Precision	Recall	Average MCC
D-VIII	0.5865	0.8711	0.8702	0.8770	0.4696
D-IX	0.6447	0.8992	0.8980	0.9015	0.4775
D-X	0.4346	0.8894	0.9157	0.8735	0.3754
D-XI	0.6449	0.8998	0.8986	0.9019	0.5935
D-XII	0.7189	0.9294	0.9311	0.9282	0.4499
D-XIII	0.5956	0.8891	0.8880	0.8905	0.5547
D-XIV	0.7234	0.9235	0.9228	0.9247	0.5737

a much better result when the areas were inpainted. We also observed that by removing the corners on the Kvasir set during training, the testing on the CVC-sets we did not get any better results in general. This was as expected since all the images had black edges, and removing them from training would make the datasets less alike. Our second hypothesis was concerning the removal of the green squares in the training set. With this, we wanted to see how the inpainted training sets affected to the test set that did not originate from the original distribution. We observed good results for both the CVC-12k set and the CVC-356 set. For the set, we deemed most realistic, namely the CVC-356 set, we saw that our score consistently was higher both for the average and the max MCC. Lastly, using a non-optimised network gives a lower MCC score when inpainting. In general, we see that inpainting to only remove sparseness will often worsen the results when the test and training set is from different sources. The same goes for excessive inpainting.

VII. Conclusions

Our two main hypotheses regarding types of inpainting for this paper were about how it would affect classification. We tested this on various datasets with different models at different sizes to see how the datasets affected the classification score. From our experiments, we can see that inpainting can help when generalising the training data to other datasets. In our GI anomaly classification experiments, our models show an average increase of at least 7% MCC score when using an optimal network for testing on images that are not from the same domain as the training data, shown in VII. When working with bigger size images, and subsequently larger areas with sparse information, it seems that inpainting does a better job, compared to smaller images. The results coincide with the previous work done [4].

References

[1] B. Stewart and C. Wild, *International Agency for Research on Cancer. World Cancer Report 2014 (International Agency for Research on Cancer)*. World Health Organization, 2014.

[2] K. Pogorelov, K. R. Randel, C. Griwodz, S. L. Eskeland, T. de Lange, D. Johansen, C. Spampinato, D.-T. Dang-Nguyen, M. Lux, P. T. Schmidt, M. Riegler, and P. Halvorsen, "Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, ser. MMSys'17. New York, NY, USA: ACM, 2017, pp. 164–169. [Online]. Available: http://doi.acm.org/10.1145/3083187.3083212

[3] K. Pogorelov, M. Riegler, P. Halvorsen, S. Hicks, K. Randel, D.-T. Dang-Nguyen, M. Lux, O. Ostroukhova, and T. Lange, "Medico multimedia task at mediaeval 2018." CEUR Workshop Proceedings (CEUR-WS.org), 2018.

[4] M. Kirkerød, V. Thambawita, M. Riegler, and P. Halvorsen, "Using preprocessing as a tool in medical image detection." CEUR Workshop Proceedings (CEUR-WS.org), 2018.

[5] S. Hicks, M. Riegler, P. Konstantin, K. V. nonsen, T. de Lange, D. Johansen, M. Jeppsson, K. R. Randel, S. Eskeland, and P. Halvorsen, "Dissecting deep neural networks for better medical image classification and classification understanding," 2018.

[6] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International Journal of Remote Sensing*, vol. 28, no. 5, pp. 823–870, 2007. [Online]. Available: https://doi.org/10.1080/01431160600746456

[7] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2016.278

[8] E. Denton, S. Gross, and R. Fergus, "Semi-supervised learning with context-conditional generative adversarial networks," 2016.

[9] S. M., "Cross-validatory choice and assessment of statistical predictions." *Journal of the Royal Statistical Society*, no. 36(2), pp. 111–147, 1974.

[10] R. J. Borgli, P. Halvorsen, M. Riegler, and H. K. Stensland, "Automatic hyperparameter optimization in keras for the mediaeval 2018 medico multimedia task." CEUR Workshop Proceedings (CEUR-WS.org), 2018.

[11] Y. K. H. Bourlard, "Auto-association by multilayer perceptrons and singular value decomposition," 1988. [Online]. Available: http://ace.cs.ohio.edu/ razvan/courses/dl6890/papers/bourlard-kamp88.pdf

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[13] S. Hicks, P. H. Smedsrud, P. Halvorsen, and M. Riegler, "Deep learning based disease detection using domain specific transfer learning." CEUR Workshop Proceedings (CEUR-WS.org), 2018.

[14] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2017.243

[15] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016.

[16] F. Chollet, "Applications - keras documentation. [online]," https://keras.io/applications/, 2019, accessed: 2019-01-07.

[17] J. Bernal and H. Aymeric, "Miccai endoscopic vision challenge polyp detection and segmentation," 2017, accessed: 2019-01-07. [Online]. Available: https://endovissub2017-giana.grand-challenge.org/home/

[18] E. L. Denton, S. Gross, and R. Fergus, "Semi-supervised learning with context-conditional generative adversarial networks," *CoRR*, vol. abs/1611.06430, 2016. [Online]. Available: http://arxiv.org/abs/1611.06430

[19] B. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," *Biochimica et Biophysica Acta (BBA) - Protein Structure*, vol. 405, no. 2, pp. 442 – 451, 1975. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0005279575901099