Detecting Slag Formations with Deep Convolutional Neural Networks

Christian von Koch[1], William Anzén[1], Max Fischer[1], and Raazesh Sainudiin[1,2,3](✉)

[1] Combient Mix AB, Stockholm, Sweden
{christian.von.koch,william.anzen,max.fischer}@combient.com
[2] Combient Competence Centre for Data Engineering Sciences, Uppsala University, Uppsala, Sweden
[3] Department of Mathematics, Uppsala University, Uppsala, Sweden
raazesh.sainudiin@math.uu.se
https://combient.com/mix, https://math.uu.se/

Abstract. We investigate the ability to detect slag formations in images from inside a Grate-Kiln system furnace with two deep convolutional neural networks. The conditions inside the furnace cause occasional obstructions of the camera view. Our approach suggests dealing with this problem by introducing a convLSTM-layer in the deep convolutional neural network. The results show that it is possible to achieve sufficient performance to automate the decision of timely countermeasures in the industrial operational setting. Furthermore, the addition of the convLSTM-layer results in fewer outlying predictions and a lower running variance of the fraction of detected slag in the image time series.

Keywords: Image segmentation · Deep neural network · Iron ore pelletising plant · Furnace slag-detection

1 Introduction

The mining industry is an ancient industry, with the first recognition of underground mining dating back to approximately 40,000 BC [5]. Historically, mining consisted of heavy labour without the aid of modern technology, but today the industry portrays a different story. Large machines and production lines are responsible for carrying out the mining process more efficiently and at scale. This is evident in the process of transforming raw iron ore into iron ore pellets – small iron ore "balls" used as the core component in steel production.

Iron ore pellets are produced in large pelletising plants, where iron ore concentrate is mixed with bentonite, a clay mineral acting as a binder. The ore and bentonite mix is then formed into pellets in a rotating drum. At the final stage of this process, the wet iron ore pellets are pre-heated and dried in a large furnace called the Grate-Kiln system. The Grate-Kiln system [14] consists of three main units: 1. Drying and pre-heating (Grate), 2. Heating (Kiln) and 3. Cooling.

When the pellets flow between the Grate and the Kiln, a common challenge in pelletising plants is that an unwanted by-product known as slag slowly builds

C. Bauckhage et al. (Eds.): DAGM GCPR 2021, LNCS 13024, pp. 559–573, 2021.
https://doi.org/10.1007/978-3-030-92659-5_36

up over time. The slag gets attached to the base (slas) of the Kiln and restricts the flow of the pellets, worsening the quality of the end-product and, if it breaks loose, it can potentially damage the equipment. Thus, it is of vital importance to track the slag build-up over time, as timely countermeasures have to be taken to remove the slag before it grows too large.

One solution used to track the build-up is to attach a video camera positioned to record the slag inside the furnace. The video stream is then manually and visually analysed by employees at the plant. While this process enables plant operators to get an instant snapshot of the current state of the slag, the process is not only labour-intensive but also prone to inconsistencies due to individual human biases. Moreover, it does not provide a historical systems-level overview of how the amount of slag has changed over time.

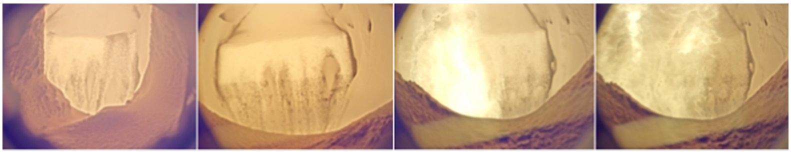


Fig. 1. Sample images with varying conditions from within the Kiln. The two images to the left are occlusion-free and the two images to the right contain occlusions.

Our aim is to research the possibility of measuring the amount of slag that has accumulated in the Kiln through image segmentation with deep convolutional neural networks (DCNNs). Furthermore, we want to assess the effects of utilising consecutive pairs of images in the time series of images when segmenting possibly occluded images (e.g. flames and smoke in the third and fourth images of Fig. 1). This will be done by introducing a convLSTM-layer in the DCNN. Our work contributes with the following novel findings:

1. We show that it is possible to measure the amount of slag that has accumulated in a Grate-Kiln system through image segmentation with DCNNs.
2. We show that it is possible to reduce variance in the predicted slag fraction by using consecutive pairs of images through a convLSTM-layer in the DCNN.
3. We demonstrate an approach for handling images containing occlusion in the video stream from the Grate-Kiln system. We believe that this approach can be applied to other applications where occasional obstructions of the camera view are considered problematic.
4. We have developed image segmentation models to predict slag formation through a scalable [24] delta lake architecture [2] and deployed in production, thus showing that our work is not only sufficient in theory, but also brings value in practise.

In Sect. 3, we describe the raw dataset and models used for conducting our experiments. In Sect. 4, we describe the first of our experiments which compares two DCNNs at the task of identifying slag formations in images. We show that it is possible to detect slag formations with both models, which to our knowledge has not been shown in the literature before. In Sect. 5, we analyse how utilising temporal information might be beneficial for lowering prediction variance and handling images with occlusions in them. Finally, we summarise our results and findings in Sect. 6.

2 Related Work

Image segmentation has been used successfully in a vast number of applications such as medical image analysis, video surveillance, scene understanding and others [17]. The most significant model and architecture contributions within the image segmentation field have been reviewed and evaluated by Minaee et al. [17], whose article covers more than 100 state-of-the-art algorithms for segmentation purposes, many of which are evaluated on image dataset benchmarks.

The use of image segmentation models in the field of mining is limited in the literature, but some research has been conducted on measuring the size of pellets and other mining-related particles through computer vision. Hamzeloo, et al. used a combination of Principal Component Analysis (PCA) and neural networks to estimate particle size distributions on industrial conveyor belts [8]. Support Vector Machines (SVMs) have been successfully used to estimate iron ore green pellet size distributions in a steel plant [10] and a DCNN (U-Net architecture) has been proposed for the same purpose [6].

To our knowledge, no research has been conducted on detecting slag formation through computer vision within furnaces used in the mining industry. A reason for this might be due to the Grate-Kiln system previously being regarded as a black-box process [26]. Gathering data through measuring devices that can withstand temperatures above 1000 °C has been difficult historically.

When evaluating videos or sequences of images, where potential insights from temporal information can be obtained, there are multiple approaches that can be used. For example, one can concatenate the current frame with the previous N frames to a common input and process these frames with an ordinary CNN. However, Karpathy et al. [12] showed that this only results in a slight improvement and that the use of a temporal model proved to be a more successful approach. Multiple authors have since then proposed the use of recurrent neural networks (RNNs) when performing image segmentation on videos or sequences of images. For example, Valipour et al. [22] proposed a modification of a Fully Convolutional Network (FCN) [15] with an added RNN unit between the encoder and decoder, achieving a higher performance on the Moving MNIST dataset [20] and other popular benchmarks. Yurdakul et al. [7] researched different combinations of convolutional and recurrent structures, finding that an additional convLSTM cell yielded the best result. Pfeuffer et al. [18] investigated where to place the convLSTM cell in the DCNN model architecture, finding that the largest increase in performance was achieved by placing it just before the final softmax layer. We use insights from such recent work on image segmentation, but with novel adaptations to our problem of slag detection above 1300 °C.

3 Raw Dataset and Models

In this section, the raw dataset and the deep convolutional neural network (DCNN) models for image segmentation used in our experiments are introduced.

3.1 Raw Dataset

All the data used in this report was supplied by LKAB (a Swedish mining company) and collected from one of their pelletising plants located in Kiruna, Sweden. The dataset at our disposal consisted of a stream or time series of 640 × 480 pixel RGB images. Each image was captured inside the Kiln with a 10-second time lag, and included metadata about when each image had been taken. The state of the furnace as well as the angle and position of the camera affect the characteristics of the images. For example, the furnace might be in its initial stages of heating causing a different lighting than when it has just been turned off. Furthermore, the camera sometimes had its position moved due to the vibrations within the furnace, causing the previous landscape to change. Lastly, the images captured sometimes contain a flame, used for heating the Kiln, with subsequent smoke. Both flame and smoke can occlude the view of the slag in the image. The images were ingested into a delta lake [2] to easily train our models.

3.2 Models

U-Net. The first model implemented in this project was the U-Net [19]. The U-Net is an early adaptation of the encoder-decoder architecture outperforming other state-of-the-art architectures of that time and field by a large margin. The U-Net produces high accuracy segmentation while only requiring a small training dataset. By utilising the feature maps of the encoder combined with the up-sampled feature maps of the decoder, localisation of features can be made in a higher resolution, thus increasing performance.

PSPNet. The second model implemented for our experiments was the Pyramid Scene Parsing Network (PSPNet) [25]. The PSPNet is a network that aims to utilise the global prior of a scene by downsampling the image into different sized sub-regions through average pooling. By doing this, both global and local information of each pixel can be exploited.

The network uses a backbone of a Residual Network (ResNet) [9] with a dilated strategy, i.e., replacing some of its convolutional operations with dilated convolutions and a modified stride setting, as described in [4,23]. ResNets are based on the idea of a residual learning framework that allows training of substantially deeper networks than previously explored [9]. Following the backbone, the architecture uses its proposed pyramid pooling module, a concatenating step, and finally a convolutional layer to create the final predicted feature maps.

4 Detecting Slag Accumulation with Deep Neural Networks

Here, we discuss the experiment with the aim of researching the potential of measuring slag accumulation using image segmentation with the U-Net and the PSPNet, two DCNN models.

4.1 Dataset

To train the DCNN models, a subset of the raw images was extracted. These images were manually and carefully chosen to ensure a high variance that is representative of the full distribution of images. Furthermore, the images chosen did not contain any occlusions to reduce interference and noise in the training process. Finally, the subset of images were manually labelled into four classes using the COCO-annotator tool [3]: *background*, *slag*, *camera edge* and *wall* (see Fig. 2). The classes were chosen based on discussions with domain experts at LKAB. *Background* and *slag* were needed to compute the percentage of slag stuck on the flat surface of the Kiln (i.e. the slas). *Camera edge* was needed to alert local plant maintenance when the camera hole needed cleaning. *Wall* simply represented the rest of the pixels. This resulted in a dataset consisting of 530 images, with 339 used for training, 85 for validation, and 106 for testing.

4.2 Implementation Details

U-Net. The U-Net implemented in our work was built as described in the original paper, with the exception of using less filters in its layers. The number of filters used was reduced by a factor of 16 in comparison to the original architecture, e.g., the first layer of the original architecture used 64 filters and our implementation used 4 filters in the same layer. The decision of using fewer filters in the implementation was made since no significant improvement in performance was observed by using the full-scale U-Net.

PSPNet. The PSPNet was built with a backbone of ResNet-50, a version of the ResNet model [9] with 50 convolutional layers. This was done by utilising the pre-built ResNet-50 model available in TensorFlow [21], with slight modifications to suit the dilated approach of the PSPNet, as described in [4,23]. The weights of the backbone ResNet-50 was initialised using pre-trained weights, trained on the ImageNet dataset.

Loss Functions and Class-Weighting Schemes. When analysing the class distribution of the dataset defined in Sect. 4.1, it became evident that some classes were under-represented, causing a *class imbalance problem*; see Fig. 2. Class imbalance within a dataset has been shown to decrease performance for some classifiers, including deep neural networks, causing the model to be more

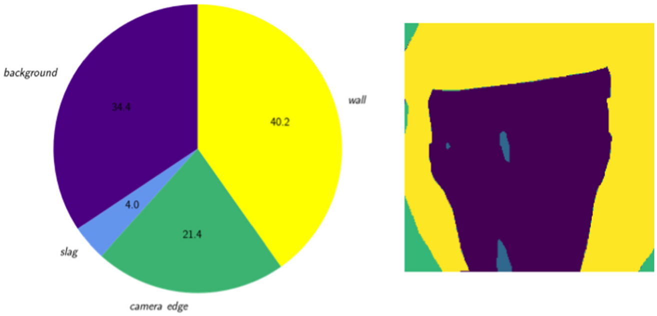


Fig. 2. Visualisation of the skewed class distribution present in the dataset used for training the models in experiment 1. The classes in the image are background (purple), slag (blue), camera edge (green) and wall (yellow). (*Left*): Class distribution of the data. (*Right*): Example ground truth image showing the skewed distribution of pixels. (Color figure online)

biased towards the more frequently seen classes [11]. To tackle this problem, we experimented with three loss functions; the *Tanimoto loss* [5], the *dice loss* [16], and the *cross-entropy loss*. We also experimented with two class-weighting schemes; the *inverse square volume weighting* (ISV) and the *inverse square root volume weighting* (ISRV), in comparison to no class-weighting scheme. The ISV weighting scheme was calculated by $w_c = 1/f_c^2$ and the ISRV weighting scheme was calculated by $w_c = 1/\sqrt{f_c}$, for each segmentation class c, where f_c denotes the frequency of pixels labelled as class c in the dataset used for training the model. The ISV weighting scheme together with the Tanimoto loss as well as the ISRV weighting scheme together with the cross-entropy loss function have been successfully used for class imbalance problems [1,5]. We found that the best performance in our trials was achieved by the dice loss with ISV and the cross-entropy loss with ISRV for the U-Net and the PSPNet models, respectively. For more details on ablation studies please see Sect. 5.2 of [13].

4.3 Evaluation Metrics

Pixel Accuracy. The most common evaluation metric used for classification problems is accuracy. For image segmentation tasks, pixel accuracy is frequently used. It is defined as the proportion of correctly classified pixels, as follows:

$$\text{accuracy} := \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}}\,, \tag{1}$$

where TP, FP, TN and FN refer to the *true positives*, *false positives*, *true negatives* and *false negatives*, respectively.

Intersection over Union. One of the most commonly used metrics for image segmentation is the *intersection over union*, denoted by IoU [17]. This metric measures the overlap of pixel classifications, making it less sensitive to class imbalances within the image, in comparison to accuracy. The equation for IoU for each class c is defined below, where A denotes the predicted segmentation map over the classified pixels and B denotes the ground truth segmentation map.

$$\mathrm{IoU}_c := \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

The IoU of the class labelled *slag* was used for validation since this was the class of interest in this problem.

Mean Intersection over Union. The *mean intersection over union* metric, denoted by mIoU, averages over the IoU metric for each of the classes and is defined as follows:

$$\mathrm{mIoU} := \frac{1}{|\mathbb{C}|} \sum_{c \in \mathbb{C}} \mathrm{IoU}_c \ , \tag{3}$$

where $|\mathbb{C}|$ is the number of classes in set $\mathbb{C}$ and IoU_c is the IoU for class c.

4.4 Results and Analysis

Table 1. Class-wise IoU_c scores of the slag detection experiment with mIoU and accuracy calculated over all classes. All metrics are computed on the test set.

Model	Class c	IoU_c(%)	mIoU(%)	accuracy(%)
PSPNet	*background*	94.06	87.72	97.32
	slag	65.33		
	camera edge	94.63		
	wall	96.85		
U-Net	*background*	92.54	85.37	96.38
	slag	62.67		
	camera edge	91.61		
	wall	94.67		

The results of this experiment are presented in Table 1. The results show that both models are able to segment the images in the test set, especially for the more frequent class labels: *background*, *camera edge* and *wall*. Both the U-Net and PSPNet are also able to segment slag at an IoU of 62.67% and 65.33%, respectively, thus showing that it is possible to utilise image segmentation when detecting slag. This was also visually verified by domain experts at LKAB, verifying that the achieved IoU of slag was sufficient in an industry setting.

The lower IoU for the class *slag* in comparison to the other classes we believe is due to the class imbalance problem, mentioned in Sect. 4.2. Another reason could be an effect of the manual labelling process, as it is sometimes difficult for the human eye to detect the slag islands due to their small size and sometimes indistinguishable edges. This is evident in the worst prediction yielded by the PSPNet visualised in Fig. 3, which also shows that even though the prediction of the model yielded a low IoU score in this case, the *amount* of slag predicted can be fairly accurate.

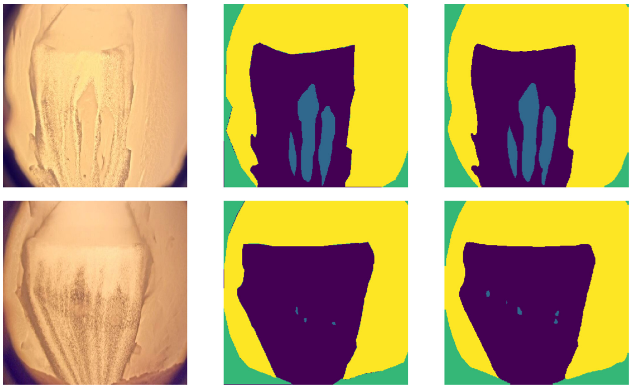


Fig. 3. Two examples of predictions yielded by the PSPNet with weighted loss. (*Top*): The best prediction, yielding 86.8% IoU of *slag*. (*Bottom*): The worst prediction, yielding 8.4% IoU of *slag*. (*Left*): The input images. (*Middle*): The ground truth images. (*Right*): The predictions made by the PSPNet.

Although both of the models showed potential in segmenting the images, the superiority of the PSPNet over the U-Net in this particular context is evident from looking at the results in Table 1, as it outperforms the U-Net in all of the metrics measured. Figure 3 displays some example predictions made by the PSPNet which we believe shows, together with the results presented in Table 1, that it is possible to measure the amount of slag that has accumulated in a Grate-Kiln system through image segmentation with a DCNN. We will in the next experiment look into whether the better performing PSPNet is consistent in its predictions by analysing its performance on unlabelled images over a longer time-frame.

5 Variance and Occlusion Experiment

Here, we present the experiment with the aim of researching the performance in a productionised setting. It includes analysing the prediction variance and the ability to deal with occluded images hindering the view of the camera.

5.1 Datasets

Training Dataset. The training dataset, used in Sect. 4, was extended by adding the subsequently captured images, captured with a 10 s time lag. This resulted in a time series dataset containing consecutive pairs of images. As the images captured in the Kiln sometimes include flames and subsequent smoke obstructing the view, the resulting dataset only contained 420 pairs of images, i.e., 110 out of the 530 subsequent images were discarded due to the presence of an occlusion. The newly added images were labelled in the same manner as was described in the first experiment; see Sect. 4.1. Out of the 420 pairs of images, 269 pairs were used for training, 67 used for validation, and 84 used for testing. After testing the performance of the models on unseen data, we retrained the models prior to evaluation on the following *evaluation datasets*, using 336 pairs for training and 84 for validation.

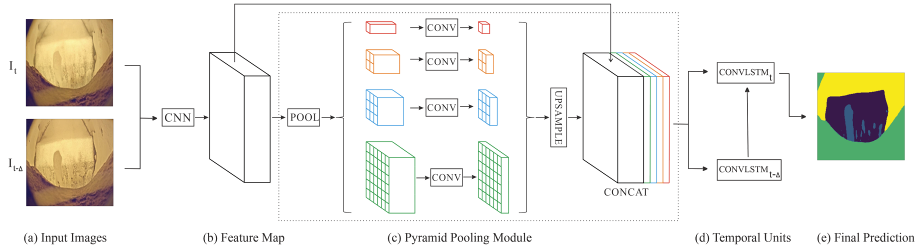
Evaluation Datasets. To evaluate the robustness of the models in a productionised setting, raw consecutive images taken in sequence from two separate days were extracted into two datasets, containing 7058 and 8632 images respectively. The two days were chosen because of their different characteristics. One of the days contained images that showed a slow and gradual slag build-up, whereas the other day contained images where it was known beforehand that slag had been removed throughout the day, resulting in a varying slag build-up. These images were not labelled.

5.2 Implementation Details

Occlusion Discriminator. In order to evaluate the DCNN models on the unlabelled evaluation datasets, the images input to the models needed to be occlusion-free. To replicate a production setting and to reduce the manual work of removing images containing occlusions, an *occlusion discriminator* model was implemented and trained. The model was a DCNN classifier of three convolutional blocks, each block containing two convolutional layers each of which was followed by a batch normalisation and ReLU activation functions. Each convolutional block was then followed by a max pooling and dropout layer. Finally, a dense layer was followed by a sigmoid function, yielding binary predictions on whether the image contains occlusions or not. After training the model, it achieved a precision and recall score of 99.7% and 83.7% respectively. This model was used to filter the evaluation datasets.

Two PSPNet Models. The best performing hyperparameters of the PSPNet from the previous experiment, described in Sect. 4, was used in this experiment. It was trained using the cross-entropy loss with the ISRV class-weighting scheme, and is referred to as the PSPNet. The PSPNet was evaluated in comparison to a modified version of the same PSPNet model, referred to as the PSPNet-LSTM. The PSPNet-LSTM was modified by replacing the final convolutional layer with a convLSTM-layer with two cells, with shared weights, to enable the model to process pairs of images. Furthermore, the layers prior to the convLSTM-layer were implemented to ensure that both of the input images were processed by the same weights.

The convLSTM-layer utilised information from a sequence of images. When predicting a segmentation mask of an image I_t of time step t, it used information from both I_t and the hidden state of the previous LSTM cell utilising information from $I_{t-\Delta}$, with Δ set to 10 s in our experiments. The output of the PSPNet-LSTM was a segmentation mask prediction of the image I_t. See a visualisation of the model in Fig. 4. In summary, these modifications enabled the PSPNet-LSTM to train on pairwise images.



(a) Input Images (b) Feature Map (c) Pyramid Pooling Module (d) Temporal Units (e) Final Prediction

Fig. 4. Visualisation of the PSPNet-LSTM, our modified PSPNet with convLSTM cells. The images I_t and $I_{t-\Delta}$, seen in (a), are both processed by the same weights in the network. The feature maps produced from I_t are fed into the convLSTM unit at time step t, and the feature maps produced from $I_{t-\Delta}$ are fed into the convLSTM unit at time step $t - \Delta$. Finally, a segmentation map prediction of the image I_t is made.

To be able to compare the performance of the two models, the PSPNet model was retrained using the images (only I_t) defined in Sect. 5.1. The weights of PSPNet-LSTM were then initialised using the frozen pre-trained weights from the PSPNet. By using the same weights in both models and only training the additional convLSTM-layer in PSPNet-LSTM (using both I_t and $I_{t-\Delta}$), a comparison between the predictive ability of the two models on the images, with and without temporal information, could be made.

5.3 Evaluation Metrics

IoU, Fraction of Classified Slag and Running Variance. The IoU of the class *slag*, defined in (2), was used for validation and testing, when training the PSPNet. Since the purpose of this experiment was to evaluate the stability of the models in a productionised setting, raw images from two days were used for evaluation, defined in Sect. 5.1. These images were not labelled, thus the IoU could not be used. As a measurement of stability over time, the fraction of classified slag was used for evaluation:

$$\mathfrak{f}_t = \frac{n_s^{(t)}}{H \times W} \tag{4}$$

where $n_s^{(t)}$ is the number of pixels classified as slag and H and W represent the dimensions of the image at time t.

The running variance was calculated for each measurement of the fraction of classified slag at time t, i.e., $\mathfrak{f}_t$, over a fixed window of size k, prior to t. We set k to 60 amounting to 600 s or 10 min of the natural operational window for counter-measures..

$$\text{Variance}(\mathfrak{f}_{t-k+1} + \mathfrak{f}_{t-k+2} + ... + \mathfrak{f}_t) \tag{5}$$

5.4 Results and Analysis

When comparing the best performing PSPNet with the best performing PSPNet-LSTM, evaluated on the unseen test data held out during training (Sect. 5.1), the PSPNet outperformed the PSPNet-LSTM with a small margin in terms of the IoU for the class *slag*, yielding 62.39% and 61.76% respectively. The models were then evaluated on the two days of images to measure their robustness in a productionised setting by using the running variance of slag fraction—a domain-specific metric that can highlight discrepancies in performance by manual examination before operational countermeasures.

Figure 5 shows the predicted fraction of slag on the raw evaluation dataset (two full days of unlabelled images) and the corresponding running variance. The figures show that the predictions yielded by the PSPNet-LSTM has fewer outliers and is more consistent compared to the PSPNet predictions. Furthermore, the running variance is consistently lower for the PSPNet-LSTM during the period when the slag is slowly accumulating. A low and stable running variance is preferred since it should resemble the slow process of slag build-up.

We believe that the reason for the more stable predictions made by the PSPNet-LSTM is its ability to produce an adequate segmentation map surrounding one occluded image where it can partly disregard an occlusion obstructing the view. Even with an occlusion discriminator in production filtering out most occlusions, false negatives are likely to occur when used over longer periods of time. Figure 6 shows an image that was wrongly classified and passed through the trained occlusion discriminator filter (Sect. 5.2). Looking at the predictions made by the two models, combined with the fewer outlying predictions of the PSPNet-LSTM in Fig. 5, we arrive at the conclusion that the PSPNet-LSTM is more robust in its predictions and manages to predict slag regions behind the smoke, which obstructs the view, even though it has not been trained on images containing occlusions.

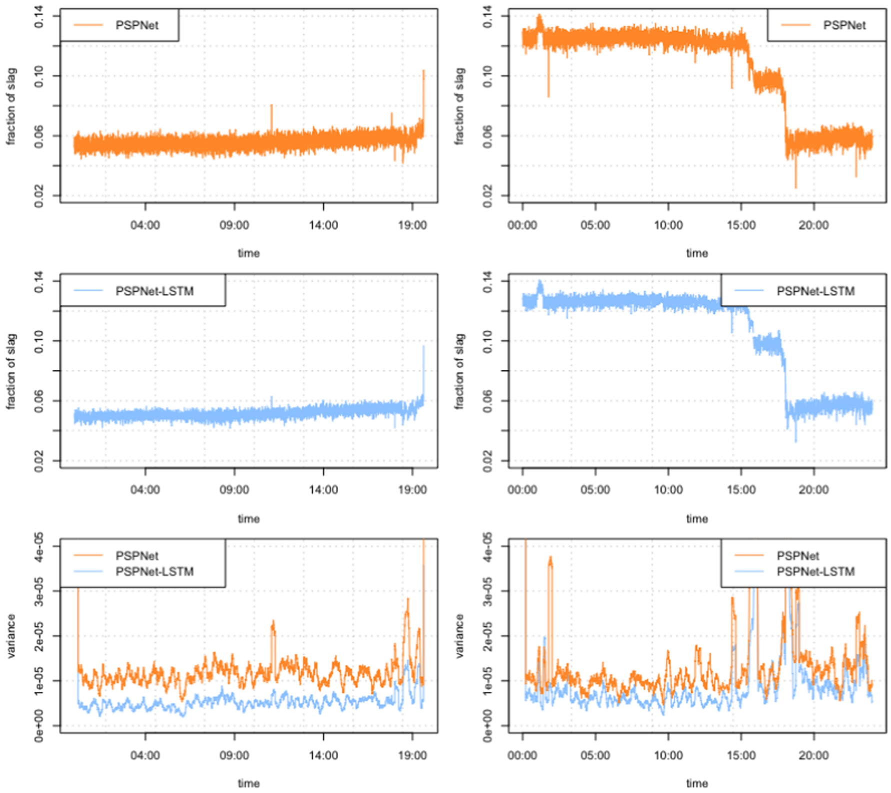


Fig. 5. Predicted fraction of slag and running variance on the evaluation dataset filtered by the occlusion discriminator.

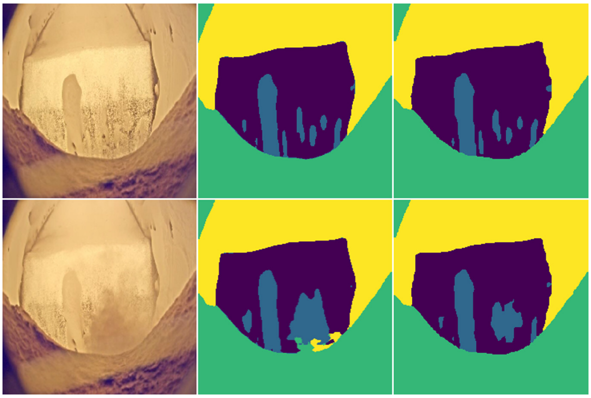


Fig. 6. (Top row): Image, $I_{t-\Delta}$, with no visible occlusion. (Bottom row): Consecutive image, I_t, with a visible occlusion that passed through the occlusion discriminator. (*Left*): The input image. (*Middle*): The prediction made by the PSPNet. (*Right*): The prediction made by the PSPNet-LSTM.

6 Conclusions

In this paper, we have researched the possibilities of measuring slag accumulation with the use of two DCNNs, the U-Net and the PSPNet. Furthermore, we have looked into how to handle images containing occlusions and proposed a novel approach to tackle this problem. We have shown that it is possible to measure slag accumulation in a Grate-Kiln system using a DCNN. The best performing model, the PSPNet, yielded an IoU of *slag* of 65.33%, an mIoU of 87.72% and an accuracy of 97.32%, all measured on test data. Finally, we have shown that a DCNN with an additional convLSTM-layer with two cells can increase the stability and lower the variance of the predictions by exploiting temporal information in consecutive images, thus making it more suitable for a production environment.

Based on interviews with domain experts at LKAB, having access to continuous measurement of slag formation will support their effort in putting in timely countermeasures to remove slag, as well as getting a better understanding of when enough slag has been removed. Additionally, having a historical and systemic view of slag formation will enable LKAB to correlate the slag build-up with other parts of the Grate-Kiln system – an important step towards understanding the root cause of the slag build-up. Such a view can be obtained by LKAB due to the raw images over multiple years being made readily available in a delta lake house to quickly build new AI models over GPU-enabled Apache Spark [24] clusters, and by integrating image data with other data sources.

Future work will focus on how to further enhance the ability of the model to deal with occluded images. We believe that this could be done by utilising longer sequences of images when making predictions. This will increase the probability of having multiple consecutive images without any occlusion, thus hopefully aiding the model in making accurate predictions. Moreover, our image delta lake architecture, over long time scales, allows to take a systems-level approach to the problem by incorporating other dependent variables.

Acknowledgements. This research was partially supported by the Wallenberg AI, Autonomous Systems and Software Program funded by Knut and Alice Wallenberg Foundation and Databricks University Alliance with AWS credits. We thank Gustav Häger and Michael Felsberg at Computer Vision Laboratory, Department of Electrical Engineering, Linköping University for their support. Many thanks to Håkan Tyni, Peter Alex, David Björnström and the rest at LKAB for answering all of our questions and supplying us with the raw data. We are grateful to Ammar Aldhahyani for the custom illustration in Fig. 4 with the kind permission of Zhao to modify the original image [25].

References

1. Aksoy, E.E., Baci, S., Cavdar, S.: Salsanet: fast road and vehicle segmentation in lidar point clouds for autonomous driving. In: 2020 IEEE Intelligent Vehicles Symposium (IV), pp. 926–932 (2020). https://doi.org/10.1109/IV47402.2020.9304694

2. Armbrust, M., et al.: Delta lake: high-performance acid table storage over cloud object stores. Proc. VLDB Endow. **13**(12), 3411–3424 (2020). https://doi.org/10.14778/3415478.3415560
3. Brooks, J.: COCO Annotator (2019). https://github.com/jsbroks/coco-annotator/
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs (2016)
5. Diakogiannis, F.I., Waldner, F., Caccetta, P., Wu, C.: Resunet-a: a deep learning framework for semantic segmentation of remotely sensed data. ISPRS J. Photogramm. Remote Sens. **162**, 94–114 (2020). https://doi.org/10.1016/j.isprsjprs.2020.01.013, https://www.sciencedirect.com/science/article/pii/S0924271620300149
6. Duan, J., Liu, X., Mau, C.: Detection and segmentation of iron ore green pellets in images using lightweight u-net deep learning network. Neural Comput. Appl. **32**, 5775 – 5790 (2020). https://doi.org/10.1007/s00521-019-04045-8, https://link.springer.com/article/10.1007/s00521-019-04045-8
7. Emre Yurdakul, E., Yemez, Y.: Semantic segmentation of rgbd videos with recurrent fully convolutional neural networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops (October 2017)
8. Hamzeloo, E., Massinaei, M., Mehrshad, N.: Estimation of particle size distribution on an industrial conveyor belt using image analysis and neural networks. Powder Technol. **261**, 185 – 190 (2014). https://doi.org/10.1016/j.powtec.2014.04.038, http://www.sciencedirect.com/science/article/pii/S0032591014003465
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016). 6
10. Heydari, M., Amirfattahi, R., Nazari, B., Rahimi, P.: An industrial image processing-based approach for estimation of iron ore green pellet size distribution. Powder Technol. **303**, 260 – 268 (2016). https://doi.org/10.1016/j.powtec.2016.09.020, http://www.sciencedirect.com/science/article/pii/S0032591016305885
11. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. Intell. Data Anal. **6**, 429–449 (2002). https://doi.org/10.3233/IDA-2002-6504
12. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2014)
13. von Koch, C., Anzén, W.: Detecting Slag Formation with Deep Learning Methods: An experimental study of different deep learning image segmentation models. Master's thesis, Linköping University (2021)
14. LKAB: Pelletizing (2020). https://www.lkab.com/en/about-lkab/from-mine-to-port/processing/pelletizing/
15. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2015)
16. Milletari, F., Navab, N., Ahmadi, S.: V-net: fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV), pp. 565–571 (2016). https://doi.org/10.1109/3DV.2016.79
17. Minaee, S., Boykov, Y.Y., Porikli, F., Plaza, A.J., Kehtarnavaz, N., Terzopoulos, D.: Image segmentation using deep learning: a survey. IEEE Trans. Pattern Anal. Mach. Intell. 1 (2021). https://doi.org/10.1109/TPAMI.2021.3059968

18. Pfeuffer, A., Schulz, K., Dietmayer, K.: Semantic segmentation of video sequences with convolutional lstms. In: 2019 IEEE Intelligent Vehicles Symposium (IV), pp. 1441–1447 (2019). https://doi.org/10.1109/IVS.2019.8813852
19. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
20. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using lstms. In: International Conference on Machine Learning, pp. 843–852 (2015)
21. TensorFlow: tf.keras.applications.resnet50.ResNet50 (2021). https://www.tensorflow.org/api_docs/python/tf/keras/applications/resnet50/ResNet50
22. Valipour, S., Siam, M., Jagersand, M., Ray, N.: Recurrent fully convolutional networks for video segmentation. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 29–36 (2017). https://doi.org/10.1109/WACV.2017.11
23. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions (2016)
24. Zaharia, M., et al.: Apache spark: a unified engine for big data processing. Commun. ACM **59**(11), 56–65 (2016). https://doi.org/10.1145/2934664
25. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
26. Zhu, D., Zhou, X., Luo, Y., Pan, J., Zhen, C., Huang, G.: Monitoring the ring formation in rotary kiln for pellet firing. In: Battle, T.P., et al. (eds.) Drying, Roasting, and Calcining of Minerals, pp. 209–216. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-48245-3_26