university of groningen / faculty of science and engineering / bernoulli institute

Fish Location Forecasting in the Norwegian Ocean using Deep Learning Techniques

Bachelor's Project Thesis

Jón Róason, s4751566, j.roason@student.rug.nl
Supervisors: Juan Diego Cárdenas-Cartagena, M.Sc.

August 31, 2024

Abstract: Successful forecasting can increase fishing efficiency by yielding larger catches, lowering fuel consumption, and reducing environmental pollution. This study aims to develop a neural network-based machine learning model capable of predicting the locations where fish will be present in the following 4 days, given the environmental features and heuristics from the previous 4 days. Datasets in matrix formats consisting of reported catch quantities, sea salinity values, and sea surface temperature are used to train the models. The U-Net model architecture is the focus of the study, as it is specialized for mapping matrices to matrices and has shown good predictive performance for tasks involving geo-spatial data. The results showed that the trained models performed poorly on test data, failing to predict future fishing locations accurately. The shortcomings of the models mainly resulted from the sparsity of the catch dataset and the inability to process temporal data natively. Despite these limitations, the study highlights the potential for future research to explore other advanced neural network architectures, such as recurrent U-Nets. Future research should also explore using more comprehensive ocean data collection methods, such as using echo-sounding data, which could mitigate sparsity, enhance model performance, and contribute to sustainable fishing practices.

1 Introduction

The fishing industry is one of the largest meat suppliers worldwide, with ocean-based food production accounting for 17% of all edible meat production. The main two sectors of food production are wild fisheries and mariculture, with wild fisheries accounting for 84% of all ocean food production in 2020. Furthermore, as the human population will increase in size throughout the century, demand for food will increase, resulting in an increased need for ocean-based food, with estimates that yields of sea-food may increase by $36 - 74\%$ (Christopher, Ling, Stefan, Á., Free, Froehlich, Golden, Ishimura, Maier, Macadam-Somer, Mangin, Melnychuk, Miyahara, De Moor, Naylor, Nøstbakken, Ojea, O'Reilly, Parma, Plantinga, Thilsted, and Lubchenco, 2020). Increasing the amount of fish caught and produced is a challenge that must be faced. However, we must also ensure we do so in a sustainable manner.

This challenge ties into the UN sustainability goals for 2030, of which 4 goals are highly relevant. These are the 2nd, 12th, 13th and 14th goals, which aim to end world hunger, ensure sustainable consumption and production patterns, combat climate change and sustainably use ocean resources (United Nations and Development, 2015). Catching more fish would work towards combating world hunger, but we must also be wary of overfishing, which has already had significant consequences. Notably, overfishing and fish habitat destruction by harmful fishing equipment have already depleted a third of worldwide fish stocks (Sumaila and Tai, 2020). As such, there is a delicate balance between supplying fish to the world's population while sustainably using ocean resources.

Furthermore, in wild fisheries, fuel consumption is an added challenge. Fishing generally involves a significant amount of searching for optimal locations where catch can be maximized, leading to more fuel consumption and decreased overall effectiveness. This results in more greenhouse gasses being released into the atmosphere, further exacerbating the effects of climate change (Kabir, Habiba, Khan, Shah, Rahim, los Rios-Escalante, Farooqi, Ali, and Shafiq, 2023).

Mitigating this trial-and-error fishing component would reduce fuel consumption, thereby decreasing pollution. One way this could be achieved is by developing methods capable of successfully predicting the fish's location before catches are carried out. Furthermore, if the quantity of fish at each location is also predicted, vessels could plan their trips before leaving so that fuel consumption is minimized. At the same time, the amount of fish caught could be more easily regulated and in line with sustainability goals.

This goal is precisely the motivation for the *FishAI: Sustainable Commercial Fishing* challenge, issued in 2022 at the *Nordic AI Meet*: develop machine learning (ML) models capable of predicting fishing locations, such that fishing vessels may plan fishing routes to increase catch effectiveness and decrease unnecessary fuel consumption (Schmidt Nordmo, Kvalsvik, Kvalsund, Hansen, and Riegler, 2022). The challenge was organized by the Norwegian Artificial Intelligence Research Consortium (NORA) and Nordic Machine Intelligence (NMI) and came with three datasets. The primary dataset consisted of historical catch notes containing where and when fish has been caught and the quantity. Additional supplementary datasets also provided in the competition consisted of sea surface temperature data, sea salinity data, and moon phase data.

This project aims to tackle this challenge in a novel way compared to previous approaches submitted to the competition. To achieve this, these existing models must first be examined so that we can identify their strengths and limitations. The next section will provide an overview of these models, their methodologies, and results.

1.1 State of the Art

Four teams who entered the FishAI competition had papers published on their approaches. These teams generally used traditional ML methods with varying success. The teams implemented a mix of classifiers and regression models. The regression models aim to predict the quantity of fish in each catch field, which is specified in the catch note dataset. The field with the highest amount of predicted fish would then be returned. Conversely, the classifiers aim to predict the catch field with the most fish directly. The primary methodologies of each paper and their performances will briefly be covered.

The first paper, *Clusters and Traveling Fishermen*, describes using random forest methods to develop regression models and classifiers. The models were tested against a naive model, which *"lists the catch field with the most fish on the analyzed day and month for each year and then takes the mode of that list"*. The authors report that neither the regression models nor the classifiers manage to outperform the naive model (Linkiö, Lahtinen, and Kolmonen, 2022).

The second paper, *The Lodestar fishing platform*, focused on implementing a regression model using XGBRegressor, a regression-specific implementation of XGBoost. Their baseline model is a simple algorithm that returns the location with the highest reported amount of fish one year before the queried date. Again, the authors report that their baseline outperformed the primary model, achieving a 67% accuracy in predicting the correct location compared to the regression model accuracy of 5% (Dammen, Brekke, Hole, Løddesøl, Roaldsnes, and Ortheden, 2022).

The third paper, *FishMAZE: Fish Monitoring and AI-Based Zone Evaluation*, implemented multiple regression models and reported that a random forest model performed best. However, they did not compare their model to any baseline and reported that their model achieved an average root-mean-squared error (RMSE) of 8.6830 (Lambon, Sagun, Saet, Maranon, and Berlin, 2022).

The fourth and final paper, *Satellite ocean data can inform precision fishing*, implements a random forest model but does not report any metrics, nor is the model compared to any baseline. However, the author states that the model did not predict locations of maximum catch (Syms, 2022).

Overall, we see a similar trend for all papers submitted to the competition, namely that they all use some tree-based model. The primary catch-note dataset is tabular, and tree-based methods perform well in such cases. However, none of the implemented models performed well, as the competition's winner was the Lodestar paper, whose baseline non-ML model performed best (Consortium, 2023).

Another area of interest is which predictive features these competing teams used. Dammen et al. (2022) did not use any of the auxiliary salinity and temperature datasets, instead relying purely on the catch notes, but which exact features were used is not specified. Linkiö et al. (2022) also did not utilize these datasets, however Lambon et al. (2022) and Syms (2022) did make use of them. Which features are relevant to this study will be further discussed in the theoretical framework section.

1.2 Contributions

The shortcomings of the ML methods used in the published FishAI challenge papers highlight the room for improvement. While all these papers have focused on traditional ML methods, this project will aim to develop better-performing models using Deep Learning (DL) methods. Specifically, this project will cover the application of U-Nets, a fully convolutional neural network architecture, to the issue of predicting optimal fishing locations.

Using the U-Net, which was initially developed for image segmentation (Ronneberger, Fischer, and

Brox, 2015), for fish location prediction is a novel approach that is promising due to its ability to capture spatial relationships present in matrices. The tabular catch note dataset will be restructured into matrices to use this ability to detect spatial relationships, an approach not taken by any published entries in the FishAI competition. Transforming the catch notes into matrices allows the catch notes and auxiliary datasets to be easily combined, as the latter are already in matrix form.

The project's research question is: **What is the performance of U-Net models for predicting the location of caught fish using historical catch data and environmental data?** By leveraging DL methods, the project aims to surpass the performance of traditional ML methods for geographical prediction problems. If successful, the implications are that the fishing industry may become more sustainable by using forecasting models for optimal route planning, minimizing the harmful environmental impact of needless fuel consumption. Within the field of AI, the approach could be generalized to other types of geographical forecasting problems that deal with multiple data types.

2 Theoretical Framework

The following section examines which features are relevant to predicting fish migration patterns. Additionally, it introduces the U-Net architecture, its applications, and how they relate to the study.

2.1 Relevance of salinity and sea surface temperature datasets

As mentioned in the introduction, not all FishAI contestants used the auxiliary salinity and temperature datasets. However, these may prove to be beneficial. The FishAI competition notes that sea surface temperature (SST) and sea salinity both influence the development of micro-algae (Schmidt Nordmo et al., 2022). This has also been further corroborated by later studies, finding that, while the effects of salinity and temperature vary from species to species, these factors can significantly influence micro-algae growth rate (Ivošević DeNardis, Novosel Vlašić, Mišić Radić, Zemła, Lekka, Demir-Yilmaz, Formosa-Dague, Levak Zorinc, Vrana, Juraić, Horvat, Žutinić, Gligora Udovič, and Gašparović, 2024) (Kholssi, Lougraimzi, and Moreno-Garrido, 2023).

As micro-algae are a primary source of nutrients for aquatic animals (Sheikhzadeh, Soltani, Heidarieh, and Ghorbani, 2024), SST and salinity may then influence the movements of fish species as they search for food. A paper from 2022 on forecasting the migration patterns of North Pacific spiny dogfish found SST correlated with migration patterns (Kanamori, Yano, Okamura, and Yagi, 2024). Specifically, the authors found that the spatial and spatio-temporal effects of (SST) influenced the migratory season of spiny dogfish. Furthermore, a paper from 2009 found that salinity may affect the movements of Bonnetheads, noting that the effect of salinity might be more pronounced during periods of significant changes in salinity levels (Ubeda, Simpfendorfer, and Heupel, 2009). This motivates our use of these auxiliary datasets.

2.2 Introduction to the U-Net Architecture

The U-Net architecture is a fully convolutional neural network (CNN) architecture, initially developed for biomedical image segmentation (Ronneberger et al., 2015). While image segmentation is the task of performing pixel-wise classification, the architecture has also been used for pixel-wise regression tasks, where it has been found to outperform state-of-the-art models (Yao, Zeng, Lian, and Tang, 2018) (Kassim, Glinskii, Glinsky, Huxley, Guidoboni, and Palaniappan, 2019). The architecture can be seen in Figure 2.1.

2.2.1 Overview of the U-Net architecture

The name "U-Net" comes from the U-shape visible in Figure 2.1. The architecture resembles encoder-decoder architectures in that inputs are progressively passed through alternating convolutional and downsampling layers, followed by alternating convolutional and upsampling layers. The encoder part of the network here is the downward pass relative to the U-shape, where the input is contracted, i.e. the spatial dimension is reduced, and the feature depth is increased. The decoder part of the network is the upward pass, where the spatial dimensions are increased again. This allows inputs and outputs to retain the same shape, facilitating the use of U-Nets for tasks mapping matrices to matrices. This report uses the term *encoder block* for a group of two convolutional layers and one max-pooling layer, and *decoder block* for a group of two convolutional layers and one transpose convolutional layer.

The differentiating factor between an encoder-decoder network and the U-Net is its use of "skip connections." These connections pass the output of each encoder block to its corresponding decoder block, where it is concatenated with the output of the previous decoder block. This permits the network to retain information from earlier points in the network and reduces the issue of vanishing gradients (Kugelman, Allman, Read, Vincent, Tong, Kalloniatis, Chen, Collins, and Alonso-Caneiro, 2022).

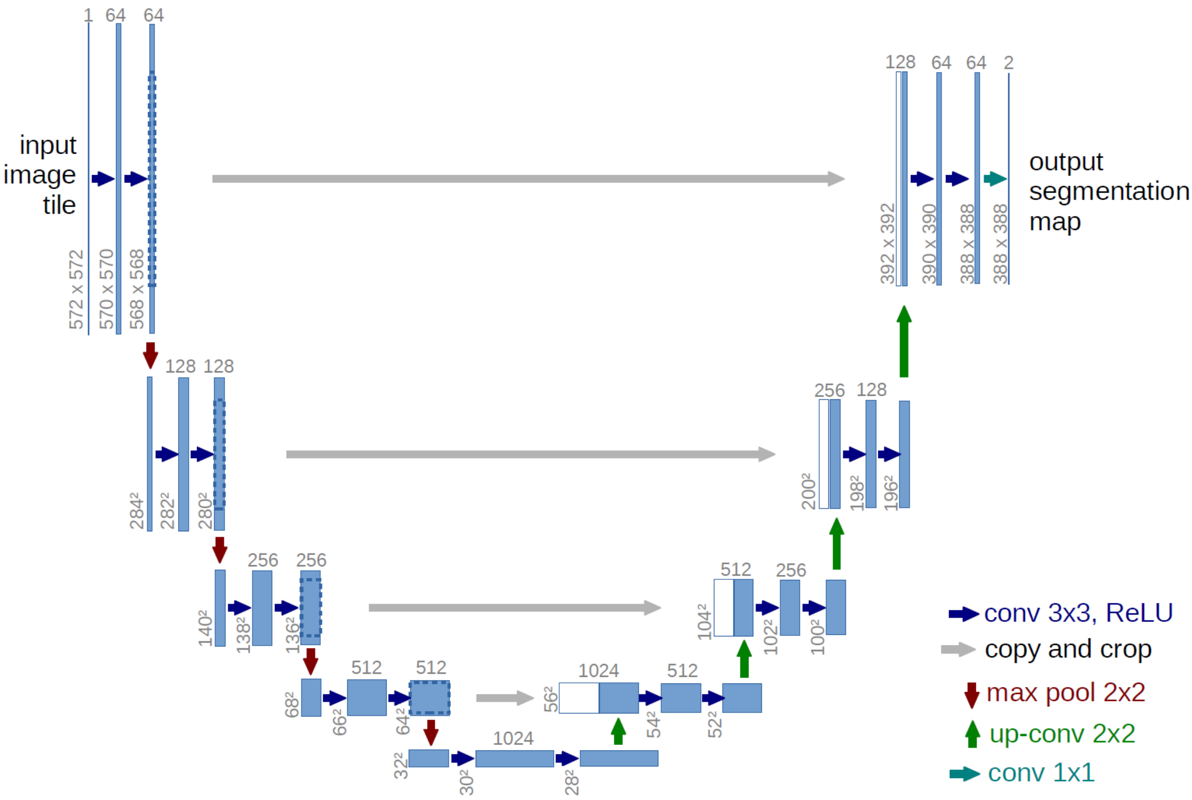

Figure 2.1: U-Net architecture overview (Ronneberger et al., 2015)

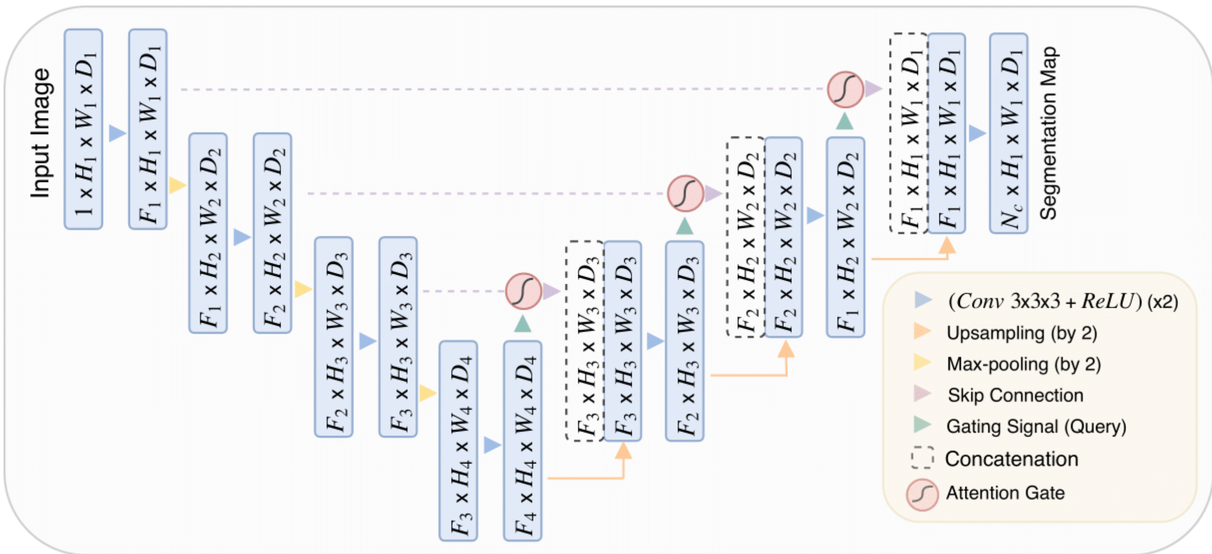

Figure 2.2: Attention U-Net architecture overview (Oktay et al., 2018)

In addition to the standard U-Net, a variant is the attention U-Net (Oktay et al., 2018). The critical difference between the standard U-Net and the attention U-Net is the use of attention gates. These attention gates process the inputs to the skip connections, as shown in Figure 2.2. The attention mechanism allows the network to focus on the spatial regions in a matrix most relevant to the output and ignore unimportant regions Kugelman et al. (2022).

In this project, both of these variants of U-Nets will be used to predict optimal fishing locations.

2.2.2 Application of the U-Net architecture for predictive fishing models

As the U-Net and its derivatives work with matrices and the primary catch note dataset is tabular, it must be converted to a matrix format. As the salinity and SST datasets are already in matrix formats, where each cell corresponds to a coordinate, the tabular data will also be transformed into this structure, with each cell containing catch weight values corresponding to the coordinate it represents. The strengths of CNNs in detecting spatial patterns can then be used, and with the attention mechanisms, the attention U-Net can identify the spatial regions most important to the output. By performing this transformation, the catch notes, SST, and salinity datasets can be combined into a higher dimensional tensor, allowing the models to account for the influence of all three variables at once.

Using the U-Net for forecasting geospatial data is a recent idea. In 2022, Fernández, Abdellaoui, and Mehrkanoon (2022) used U-Net architecture variants to predict future values of various coastal sea elements. These include sea salinity, ocean surface height, and water velocity. Additionally, Trebing, Stańczyk, and Mehrkanoon (2021) used an attention U-Net for precipitation nowcasting, while Yang and Mehrkanoon (2022) used an architecture combining transformers with U-Nets for precipitation nowcasting and cloud coverage forecasting. All these papers outperformed the baseline persistence algorithm, which uses the last image in a series as the predicted output. Lastly, Aurora, a model developed by researchers at Microsoft, uses a transformer U-Net at its core. Aurora is capable of forecasting a wide variety of atmospheric data, including air pollution levels, wind velocities, and air pressure levels (Bodnar, Bruinsma, Lucic, Stanley, Brandstetter, Garvan, Riechert, Weyn, Dong, Vaughan, Gupta, Tambiratnam, Archibald, Heider, Welling, Turner, and Perdikaris, 2024). These examples highlight the applicability of the U-Net and its variants for geospatial modelling.

2.3 Summary

The aim is to implement models that can learn to use the spatial patterns present in the transformed catch notes, salinity, and SST data to predict optimal fishing locations using the U-Net and attention U-Net models. This is a novel approach to the FishAI challenge and a potential improvement over existing, more traditional ML methods. The following section will cover the methodologies used to implement and evaluate the models, including data pre-processing steps, model design, and evaluation metrics.

3 Methods

The following section will cover the methodologies used to implement and evaluate the models, including data pre-processing steps, model design, and evaluation metrics.

3.1 Data

The primary data used are catch notes, containing a plethora of information relating to fish sold in Norway from 2000 to 2022 (Schmidt Nordmo et al., 2022), which is made available by the Norwegian Directorate of Fisheries. Auxiliary datasets used are sea surface temperature data, published by the National Oceanic and Atmospheric Administration (Reynolds, Smith, Liu, Chelton, Casey, and Schlax, 2007), and sea salinity data, published by NASA (Remote Sensing Systems, 2019).

Catch notes refer to the sales slips filled out when caught fish is landed at a landing station. These slips contain information relevant to the catch, including information about the fishing vessel, the location of the landing station, and information about the catch itself. *Sea surface temperature* (SST) and *salinity* data consist of temperature measurements obtained via satellite. The SST data consists of daily measurements from 1982 to the present day, whereas the salinity data consists of monthly averages from April 2015 to the present day. The SST and salinity datasets consist of matrices, whereas the catch note dataset is tabular and has an extensive list of approximately 130 data fields (Schmidt Nordmo et al., 2022). Therefore, the data must be inspected for the most important variables relevant to the project.

3.2 Data inspection

The most critical data variables in the catch notes dataset are shown in Table 3.1. The names of the variables are listed in their original Norwegian names, as they appear in the dataset.

Table 3.1: Relevant data in the catch note dataset.

Data type	Variable name
Location	Lat (lokasjon)
	Lon (lokasjon)
	Lat (hovedområde)
	Lon (hovedområde)
	Hovedområde
Catch weight	Produktvekt
Fish species	Art - gruppe
Catch date	Siste fangstdato

The location data contains five variables of interest, as named in Table 3.1. Four of these variables contain coordinate data; however, the contents of these variables differ in completeness and specificity. The "Lat/Lon (hovedområde)" and "Hovedområde" variables specify the 'main area' in which the landed fish was caught, whereas the "Lat/Lon (lokasjon)" variables are meant to specify more specific coordinates inside this main area at which the fish was

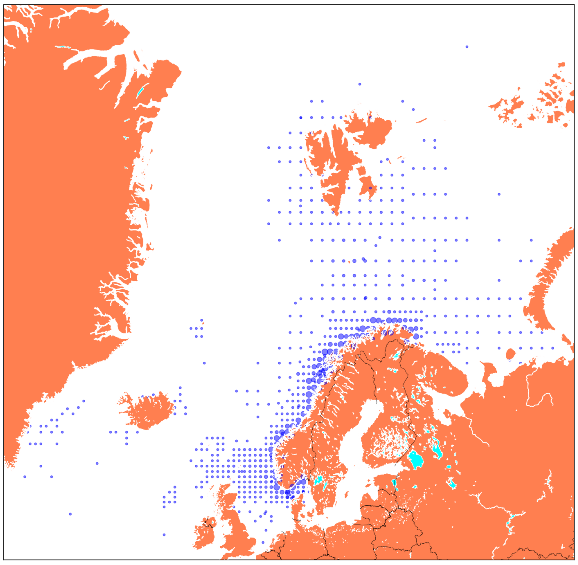

Figure 3.1: All reported catch coordinates from the training set of the catch note dataset.

caught. An inspection of the data reveals that while the "Hovedområde" variables are completely filled out, values are missing from the more specific "Lat/Lon (lokasjon)" variable. All other variables in the table do not contain any missing values. The "produktvekt" variable contains the net weight of the landed fish in kilograms. "Art - gruppe" denotes the species of the landed fish, grouped into general species categories, such as "cod." The last variable, "Siste fangstdato," denotes the last date of catch for the trip in which the landed fish was caught. Notably, this means that the date does not refer to the exact day the fish was caught but when the fishing vessel was last actively fishing.

While the temporal resolution of the SST and salinity data differ, both datasets have the same spatial resolution. Namely, the data consists of matrices, where the dimensions correspond to latitude and longitude. The dimensionality of the matrices is 720x1440 for latitude and longitude, respectively, resulting in a spatial resolution of 0.25 deg per value. Data inspection revealed noticeable missing values. Firstly, the month of July 2019 is not present in the salinity data dataset. Secondly, both datasets contain missing values on specific ocean coordinates, possibly due to the spatial resolution needing to be finer. The salinity dataset has missing values, especially near country coastlines and Arctic regions. These missing values must be filled in, as most reported catch coordinates are near land. Figure 3.1 shows the span of the coordinates in the training data split of the catch note dataset.

3.3 Data pre-processing

To avoid using extrapolated salinity values for most of our training data, only data within the interval specified by the salinity data is used, i.e. from April 2015 to January 2022. The catch note dataset within this period was split into training, validation, and test sets, using a 70/15/15 split.

As the methods used in this project have been developed for processing matrix data, where the spatial structure is relevant, the catch note dataset will be transformed from tabular data into matrices. As the salinity and sea surface temperature datasets are already in matrix form, where each cell corresponds to a geographical location, the same spatial resolution of 0.25 deg per cell will be used for the catch matrices.

An examination of the coordinates in the catch notes revealed that some were on land. These were removed from the dataset. Additionally, the catch locations reported in the dataset varied from the southern tip of Greenland to Northern Russia. Using all locations would result in very sparse matrices spanning a huge area, leading to low-frequency coordinates with less than 100 reports being removed from the dataset. The remaining coordinates fit within the intervals $[50.25, 82]$ for latitudes and $[1.75, 33.5]$ for longitudes, as shown on Figure 3.2.

The resulting catch note dataset thus consisted of 128×128 matrices for each day in the dataset, where the cells contain values representing the amount of fish caught on the coordinate corresponding to the cell. The values in the salinity and sea surface temperature data corresponding to this region were then retrieved, and the three maps were collected into one three-dimensional tensor, where one such tensor corresponds to one day. As the salinity dataset contained monthly averages, these matrices were repeated for every day in the month they were measured in, i.e. so every 3-dimensional tensor representing a day in a certain month had the same salinity values. Only two days in the used time span were not present in the catch note dataset. In these cases, zero values were filled in every cell for the catch matrices.

As the model does not have any recurrent connections, the temporal nature of the data had to be represented spatially. This is because if we were to take, e.g., two 3-dimensional tensors and combine them into a larger 4 dimensional tensor, then pass it to the network, each day would be processed independently of each other. If we input two days and output two days, then the $(n-2)$th day maps to the nth day and the $(n-1)$th day to the $(n+1)$th day, without considering the surrounding days in its predictions. That is, the model could not use the temporal order in the data for its predictions.

A window function was passed over the dataset, extracting 4 consecutive days as the input and

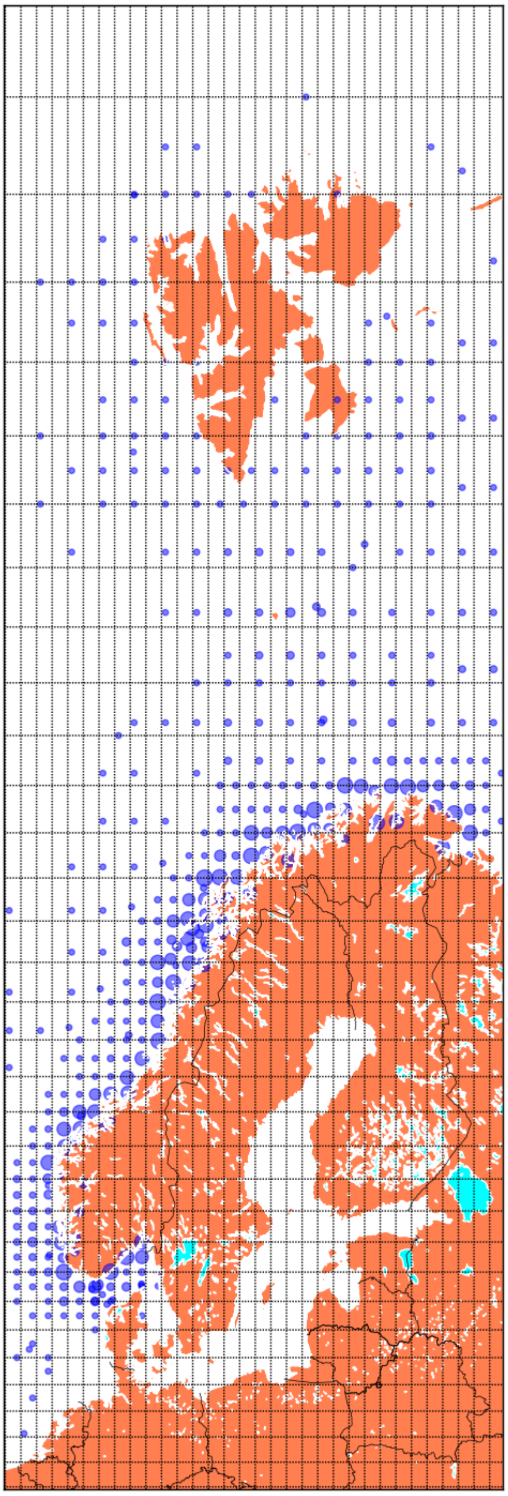

Figure 3.2: The $128x128$ bounding box containing the used coordinates. Note that the figure is not a square due to the Mercator projection being used.

the subsequent 4 consecutive days as the target. Then, each group of 4 days were spatially arranged in either a block diagonal layout or a "quadrant" layout. These layouts are visible in the matrices below, with the block diagonal layout on the left and the quadrant on the right. Here, D_i refers to the temporal order of the days. Again, this reshaping was performed for the catch, salinity and SST matrices, and then these matrices were combined into a three-dimensional tensor. This will allow the network to learn the temporal information spatially, eliminating the need for recurrent connections.

$$\begin{bmatrix} \mathbf{D}_{-3} & 0 & 0 & 0 \\ 0 & \mathbf{D}_{-2} & 0 & 0 \\ 0 & 0 & \mathbf{D}_{-1} & 0 \\ 0 & 0 & 0 & \mathbf{D}_0 \end{bmatrix} \begin{bmatrix} \mathbf{D}_{-3} & \mathbf{D}_{-2} \\ \mathbf{D}_{-1} & \mathbf{D}_0 \end{bmatrix}$$

Before training, the dataset was normalized to the range $[0, 1]$. For the salinity and sea surface temperature data, missing values were filled using mean values within the selected regions, except for the cells corresponding to land coordinates, which were left at 0. In this project, two tasks will be evaluated. The former is the task thus far discussed, namely predicting the amount of fish at each location, i.e. a regression task. The latter task is merely predicting the presence of fish in the future, i.e. a classification task. This was done to examine whether the performance of the U-Net is more robust for tasks that resemble its original purpose, segmentation tasks. Figure 3.3 shows how the targets for these two tasks differ. For the classification task, the only added processing step is converting all non-zero values in the targets to the value 1.

3.4 Model Design

All models in this study were implemented using PyTorch (Paszke, Gross, Massa, Lerer, Bradbury, Chanan, Killeen, Lin, Gimelshein, Antiga, Desmaison, Kopf, Yang, DeVito, Raison, Tejani, Chilamkurthy, Steiner, Fang, Bai, and Chintala, 2019). All U-Net models in this study contained 4 encoder and decoder, each in addition to the final convolutional layer. The U-Net models all took four consecutive days as input and predicted the immediate next four days.

Each encoder block consists of two consecutive 2D-convolutional layers followed by a max-pooling layer, while the decoder blocks consist of two consecutive 2D-convolutional layers followed by a transpose convolutional layer. The activation function of all convolutional layers was the rectified linear unit function (ReLU). Each encoder block doubled the number of feature maps of the input, with the first encoder block outputting 64 feature maps, following the design of the original U-Net.

Two versions of the U-Net models were implemented, one with and one without dropout layers. For the version with dropout, the dropout layers were placed before each pair of convolutional layers,

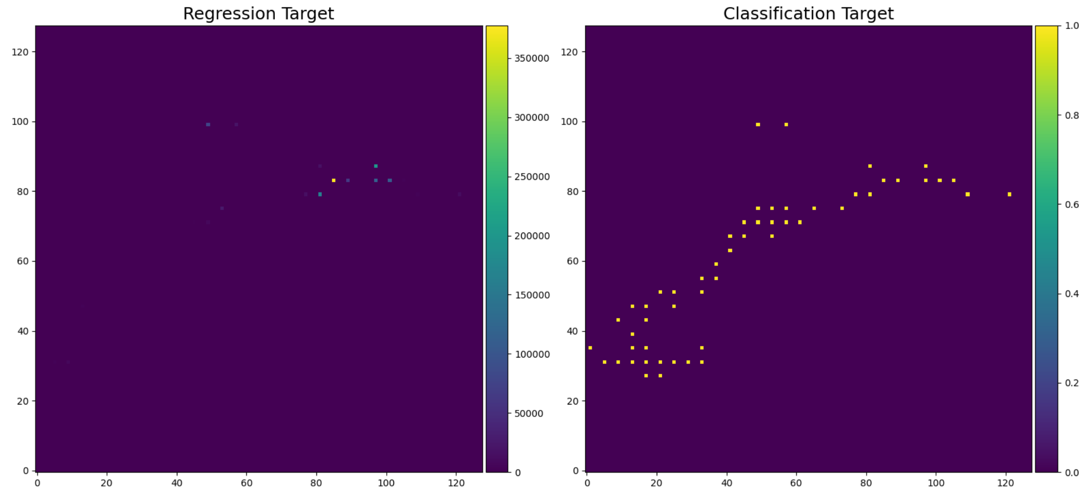

Figure 3.3: Example matrices derived from the catch note datasets. The left image shows a matrix containing the reported catch weight, whereas the right image shows the reported catch locations as a binary variable.

except the very first two such layers. Additionally, a version of the U-Net with attention gates was implemented. As the attention gates proposed by (Oktay et al., 2018) are not a part of the PyTorch package, these were implemented manually. Figure 3.4 shows the structure of these attention gates. Note that their application used 3D-convolutions since they were working with 3D-data, whereas this project used 2D-convolutions wherever convolution operations are in the diagram.

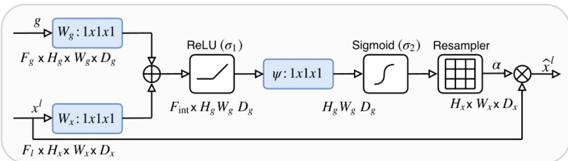

Figure 3.4: Attention gate proposed by (Oktay et al., 2018).

Thus, four different versions of the regression U-Net models were trained for each combination of dropout layers and attention gates. These models were trained using L1-loss, as the catch-note feature matrices are very sparse.

Additionally, four segmentation U-Net models were trained to predict only the presence of fish and cover each combination of dropout and attention gates. For these classification models, the inputs were the same, but all non-zero values in the target catch weight maps were set to 1. The models were trained using binary cross-entropy loss, as we had two classes to predict (0 and 1).

Finally, two final regression models were trained consisting only of the encoder portion of the U-Net; essentially, this is a standard CNN. These models contained two encoder blocks each. The CNNs differed only in the use of dropout layers. The inputs to these CNNs were the same as the others, four days, but the targets of these networks were only the next day following the input four days. The reason for this is that the output of the CNNs did not preserve the size of the input, but it was possible to achieve the size of the catch note matrices as the output shape. Training these CNNs and the U-Net models allows us to compare their performances.

3.5 Metrics

The primary evaluation metric for the regression models is mean absolute error (MAE), which is also the loss function used to train the network. For the classification models, the evaluation metric will be F1-scores and accuracy. F1-scores will probably be a better performance indicator, as the catch data predominantly contains negatives, which will likely skew the accuracy.

The Adam optimizer was used for all models' training. The regression models were trained at an upper limit of 30 epochs, and the classification models for 10 epochs. The difference in epochs was due to time constraints. Early stopping was used to combat overfitting. Additionally, the model with the lowest validation loss throughout the training was saved and used for evaluation. The batch size used for training was 8, and the learning rate was fixed at 0.001.

No hyperparameter tuning was performed, due to the large amount of models that were trained. The training time varied depending on how the data was organized, i.e. block diagonals or quadrants, and the time also differed between the regression and classification models. Two models were trained for each model parameter combination, one for each matrix organization. Table 3.2 shows how long each pair of two such models took to train.

Table 3.2: Training times for each model setup. The training is the total time it took to train two models, one for each matrix organization. The middle line separates the regression models above and the classification models below.

Model type	Training time (HH:ss)
U-Net	1 : 16
U-Net w/ attention	1 : 29
U-Net w/ dropout	1 : 18
U-Net w/ both	1 : 31
CNN	0 : 18
CNN w/ dropout	0 : 19
U-Net	0 : 26
U-Net w/ attention	0 : 30
U-Net w/ dropout	0 : 26
U-Net w/ both	0 : 30

4 Results

This section will present the performance of the regression and classification models and some examples of their predictions.

All models were trained for the number of specified epochs, i.e., no early stopping occurred. Figures 4.3, 4.4, and 4.5 show the training and validation losses of the models. These plots reveal the reason the early stopping did not occur: the validation losses of most models started oscillating as training went on, preventing the early stopping callback from executing. Another interesting feature is that the validation and training losses appear more stable in the diagonal matrix setup compared to the quadrant setup. However, the reason for this difference is unclear. Tables 4.1 and 4.2 show the evaluation metrics of the regression and classification models, respectively.*

As can be seen in Table 4.1, the un-normalized MAE is relatively high for all models. This results from the extensive range of values in the catch notes, such that even very small predicted values become large once un-normalized. This range is $[0, 2246997]$. For the classification models, we see that most of them achieve extremely high accuracy values but, conversely, very low F1 scores. This is due to the sparsity of the data, i.e., the models correctly predict a large amount of true negative 0 values but fail to predict the positive values accurately. We note that some models achieve F1-scores of 0, meaning they fail to predict any positive values correctly.

These values are not very informative, especially as we have no reference point. However, based on the metrics alone, the U-Net trained with dropout using the quadrant matrix setup has the lowest MAE score of the regression models. The U-Net using an attention mechanism and the diagonal matrix setup has the highest F1-scores of the classification models. Figures 4.1 and 4.2 show two examples of the predicted outputs of these two models, along with the ground truth values. Note that the positive targets predicted by the classification model resemble the truth values. The regression output, however, appears to be a single constant value output covering nearly the entire map. The shape of Norway is visible in the regression output, as all land values have a value of 0, but this is by design, as all land values have been masked from the predictions. The following section will discuss the performances of these models and their apparent shortcomings.

5 Conclusions

This project examined the performance of deep learning models for fish location forecasting. This section will discuss the project's key findings, the limitations of the implemented models, and ideas for future research.

5.1 Key findings

The performance of the U-Net models, as well as the CNNs, was relatively poor. The output of the regression models predominantly had constant values repeated over the entire output matrices. This pattern can be seen in the output of the dropout U-Net in Figure 4.1, as the exact value is present in every cell, except for some noise in the top right corner of the image. Therefore, these regression U-Net models without advanced modifications, as discussed in the theoretical framework section, appear infeasible for forecasting optimal fishing locations, at least with this dataset.

The classification models also showed the same pattern, as evident by some of these achieving F1-scores of 0.0, meaning that 0 values were predicted in every cell - the models showed no leaning. The best-performing model, the attention U-Net trained on diagonal input matrices, does demonstrate some learning; it has managed to predict the presence of fish along the coastline of Norway. This is not unexpected, as data inspection revealed the vast majority of reported catch locations to be within this region. However, further inspection of the

*The code for the models, data preparation, training, and prediction are available at the following Github repository: `https://github.com/JRoason/bachelors-project-RUG`

Table 4.1: MAE for regression models. Models above the middle line used the diagonal matrix format, and below the line used the quadrant matrix format.

Model type	MAE	Un-normalized MAE
U-Net	1.07×10^{-4}	180.1
U-Net w/ attention	2.7×10^{-5}	76.89
U-Net w/ dropout	2.84×10^{-5}	78.82
U-Net w/ both	5.35×10^{-5}	110.8
CNN	4.49×10^{-5}	78.98
CNN w/ dropout	3.49×10^{-5}	66.18
U-Net	3.21×10^{-5}	62.46
U-Net w/ attention	2.65×10^{-5}	55.19
U-Net w/ dropout	2.27×10^{-5}	50.23
U-Net w/ both	2.47×10^{-5}	52.9
CNN	1.6×10^{-4}	227.25
CNN w/ dropout	4.1×10^{-5}	74.03

Table 4.2: Mean test loss, F1-scores and accuracies for classification models. Models above the middle line used the diagonal matrix format, and below the line used the quadrant matrix format.

Model type	BCE Loss	F1-Score	Accuracy
U-Net	0.69	0.02	0.99
U-Net w/ attention	0.69	0.34	0.99
U-Net w/ dropout	0.69	0.00	0.99
U-Net w/ both 0.69	0.00	0.99	
U-Net	0.69	0.00	0.99
U-Net w/ attention	0.69	0.00	0.99
U-Net w/ dropout	0.69	0.00	0.99
U-Net w/ both	0.69	1.7×10^{-3}	0.97

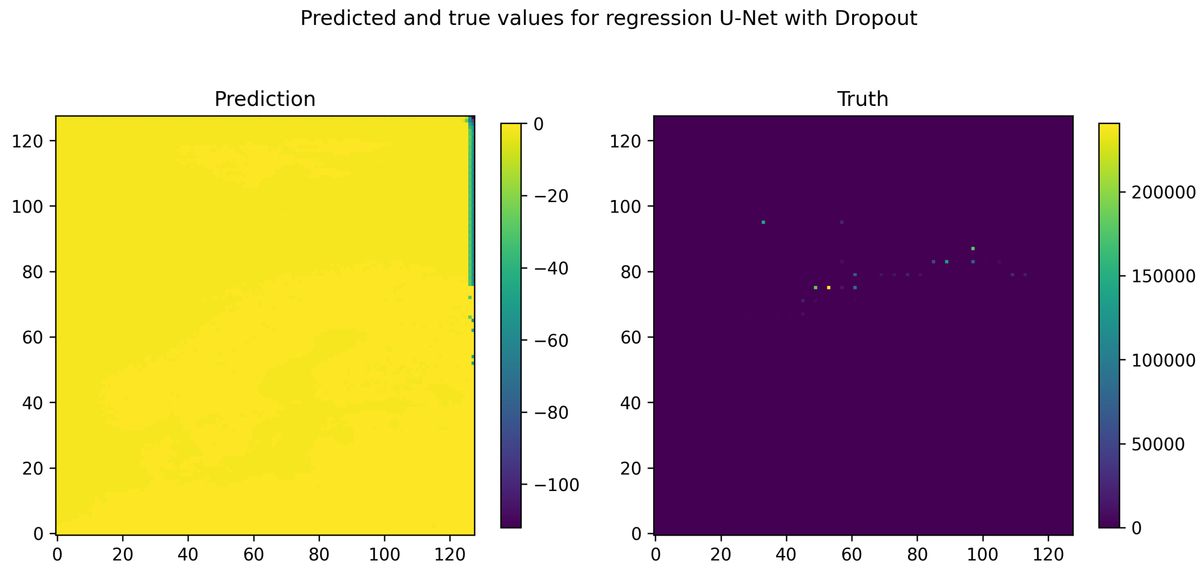

Figure 4.1: Prediction and the truth value of the regression dropout U-Net, trained on quadrant matrix setups.

predictions made by the model showed that it had overfit these locations. The model consistently predicts these same locations, regardless of the input. Some outputs differ, but only for a handful

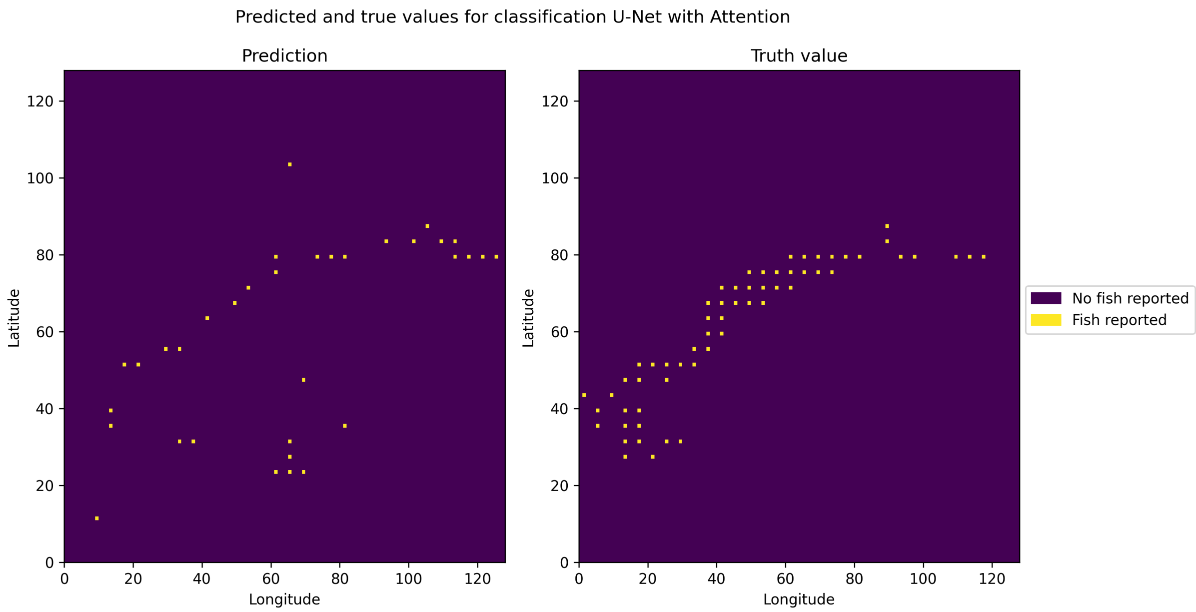

Figure 4.2: Prediction and the truth value of the classification attention U-Net, trained on diagonal matrix setups.

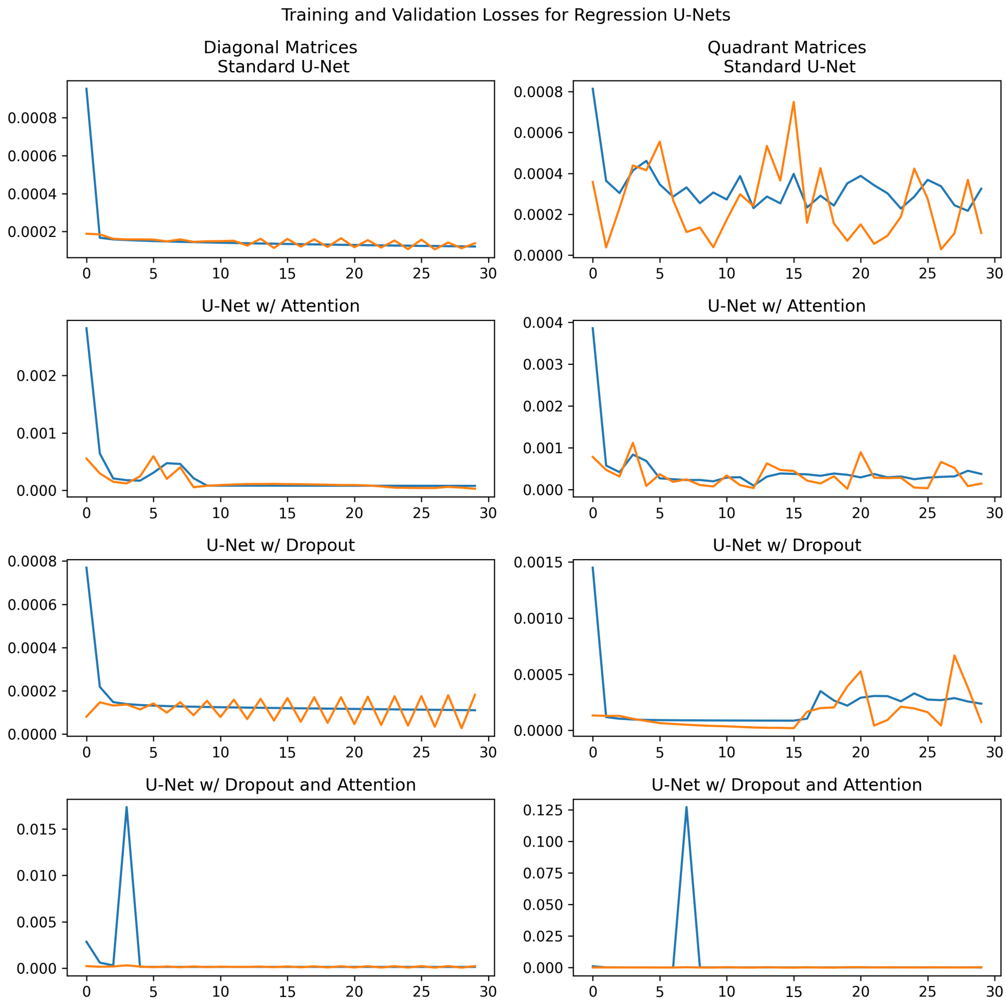

Figure 4.3: Training and validation losses of the regression U-Net models.

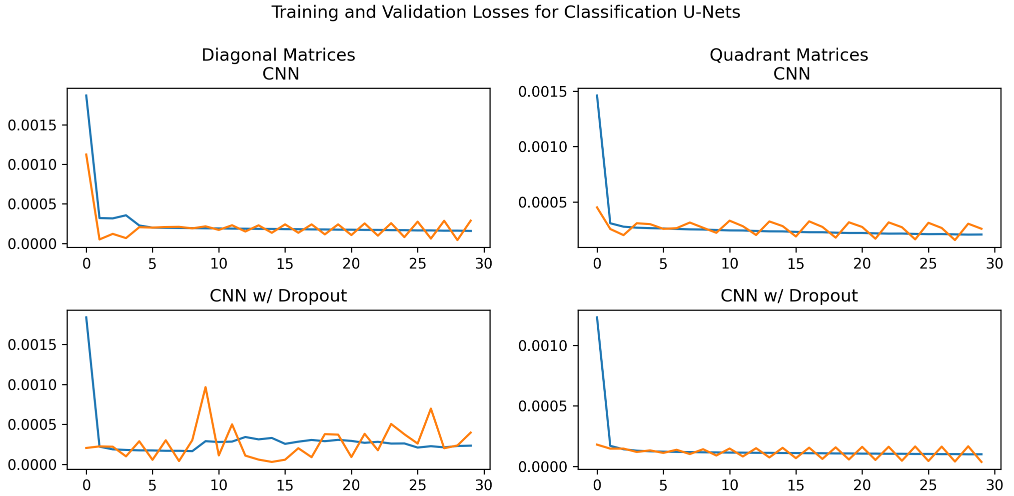

Figure 4.4: Training and validation losses of the regression CNN models.

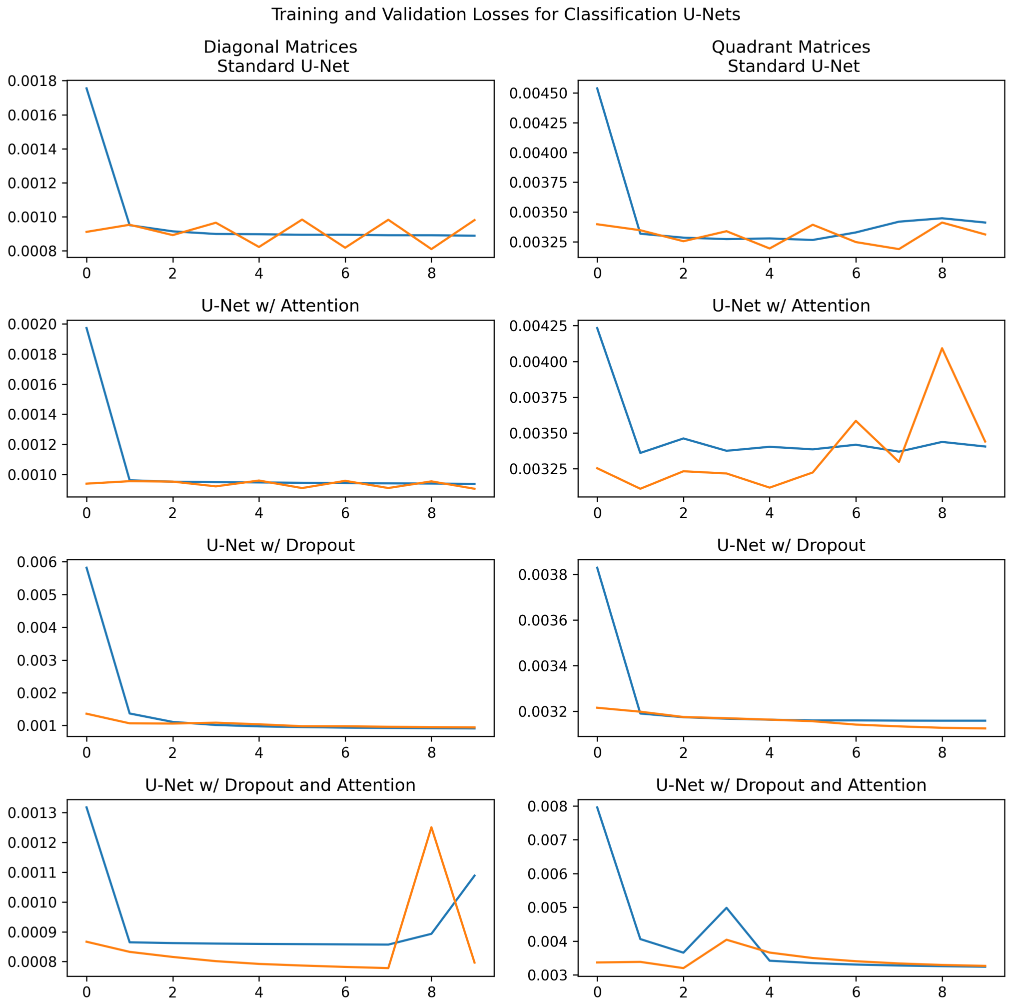

Figure 4.5: Training and validation losses of the classification U-Net models.

of cells - the shape of Norway's coastline is present in all outputs, as seen in 4.2. Further examples of this can be seen in Appendix A.

Interestingly, the attention U-Nets with dropout performed worse than this model, but not by a large amount. This could indicating that some other form of regularization might be needed, as the attention U-Net still performed better than the standard model U-Nets. This indicates that knowing which parts of the features to pay attention to when upsampling is beneficial, even though the final performances were lacklustre.

Hyperparameter tuning was not performed in this study due to time constraints. However, it may have been beneficial to perform hyperparameter tuning on the best-performing models to improve their performances. Figures 4.3, 4.4, and 4.5 show oscillations in the validation loss for all models as training went on. Stopping the training earlier or using an adaptive learning rate schedule might mitigate this issue, as it is most likely due to the learning rate being too high at later epochs.

The choice of quadrant or diagonal matrix setups to represent the temporal dimension of the data spatially affected the performance. The quadrant-trained models generally showed lower MAEs than the diagonally trained models; however, this did not result in better predictions. Instead, the quadrant-trained models were more prone to predicting the same value in all output cells. The predictions of the models trained with diagonal matrices did not make any more sense. Some of these models also output constant values, while others, such as the attention regression U-Net, learned to output constant "rotations" of values. This can be seen in Appendix A, along with other sample outputs of the models.

5.2 Limitations

A limitation of the implemented models is the need for recurrent connections. As a result, the transformation of quadrant and diagonal matrices had to be performed. The models' feedforward nature may prevent them from adequately learning the correct temporal structure in the dataset.

Another limitation is the sparsity of the transformed catch note matrices. While the original tabular version of the catch note dataset contained many entries, the matrices were sparse, mainly because we only used data from one fish species. This means that some matrices will have one or two non-zero values out of 16384 cells, where the values are usually minimal, as the maximum value in the training set was quite large. In retrospect, these extreme values ought to have been removed, which might have resulted in better performances and a more diverse range of values in the training data.

The content of the catch note dataset is in itself a limitation of the project. While the data shows the amounts of fish and where they were caught, it is essential to note that the dataset contains only true positives. The project's ultimate goal was to develop methods capable of predicting locations that would contain fish in the future; however, the target variable in this dataset is the locations where fishing vessels that have landed their catches in Norway have decided to fish. We are essentially trying to predict which locations humans have guessed to contain the most fish. Ship captains presumably have a lot of experience and knowledge regarding the best locations to fish. However, the point remains that the dataset treats all non-reported locations as having no fish, which cannot be accurate.

Lastly, the salinity dataset spans shorter than the SST and catch note dataset, leading us to limit the dataset to the period in which the salinity data exists. Effectively, we used roughly 30% of the total SST and catch note dataset, which may have made training more difficult for the models.

5.3 Future research

The U-Nets limitation of not incorporating temporal information natively encourages using architectures designed to account for ordered data for this problem. Variants of the U-Net have been developed that use long short-term memory (LSTM) layers to process temporal information (Yin, Wang, Li, Lu, Tian, Yin, Li, and Zheng, 2023), which might be more suitable than the standard U-Net. Given that attention mechanisms showed marginal improvement over the models not using attention mechanisms, this might suggest that architectures incorporating attention are promising alternatives to the models implemented in this project. Models such as the Aurora model by Microsoft, which merges transformer architecture with U-Nets might also be applicable to this problem, as they incorporate attention mechanisms and account for positional ordering in data. Another model that matches this description is the vision transformer (ViT) (Dosovitskiy, Beyer, Kolesnikov, Weissenborn, Zhai, Unterthiner, Dehghani, Minderer, Heigold, Gelly, Uszkoreit, and Houlsby, 2021). In recent years, ViTs have been used for video generation, i.e., next-frame prediction, a task involving mapping matrices to matrices. ViTs have performed comparably to state-of-the-art models (Ye and Bilodeau, 2022). The application of such network architectures to the problem of fish location forecasting warrants studying, as they might outperform the standard feedforward U-Net architecture due to incorporating temporal information directly.

Studies aiming to develop methods for fish location forecasting might achieve much greater success from datasets containing measurements of fish presence directly from the ocean. Echo-sounders are a technology used by fishing vessels to assist in choosing the best fishing locations. Access to this data might be beneficial. If such measurements were taken at many locations, and matrices were created with them, as in this project, then this would also solve the matrix sparsity problem. There is already interest in using echo-sounding data for the purpose of fish stock assessment using machine learning, leading Ordoñez, Utseth, Brautaset, Korneliussen, and Handegard (2022) to develop a standardized pre-processing pipeline for preparing echo-sounding data for machine learning models. This raises the question of how the U-Net and its derived, more advanced models might perform given more direct measurements of fish presence.

In conclusion, this study examined the performance of U-Net and convolutional neural network architectures for the problem of fishing location forecasting. The models implemented generally struggled with overfitting and poor generalization to unseen data, highlighting the complexities of modelling sparse spatiotemporal data. Future work could aim to improve the data representations and implement more advanced network architectures, such as vision transformers or recurrent U-Nets, with integrated mechanisms for processing temporal data. The main limitation, the sparsity of the datasets, could be mitigated through more accurate measurements of fish populations in the ocean without relying on catch data. Better, more informative data could lead to better results with deep learning methods developed for image processing, resulting in more accurate and sustainable fishing practices and, hence, the preservation of marine ecosystems.

References

Cristian Bodnar, Wessel P. Bruinsma, Ana Lucic, Megan Stanley, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan Weyn, Haiyu Dong, Anna Vaughan, Jayesh K. Gupta, Kit Tambiratnam, Alex Archibald, Elizabeth Heider, Max Welling, Richard E. Turner, and Paris Perdikaris. Aurora: A foundation model of the atmosphere, 2024. URL `https://arxiv.org/abs/2405.13063`.

Costello. Christopher, Cao. Ling, Gelcich. Stefan, Cisneros-Mata. Miguel Á., Christopher M. Free, Halley E. Froehlich, Christopher D. Golden, Gakushi Ishimura, Jason Maier, Ilan Macadam-Somer, Tracey Mangin, Michael C. Melnychuk, Masanori Miyahara, Carryn L. De Moor, Rosamond Naylor, Linda Nøstbakken, Elena Ojea, Erin O'Reilly, Ana M. Parma, Andrew J. Plantinga, Shakuntala H. Thilsted, and Jane Lubchenco. The future of food from the sea. *Nature*, 588(7836):95–100, dec 2020. ISSN 0028-0836, 1476-4687. doi:10.1038/s41586-020-2616-y. URL `https://www.nature.com/articles/s41586-020-2616-y`.

Nordic Artificial Intelligence Research Consortium. Meet the winners of fishai: Sustainable commercial fishing competition 2022 - nora, December 2023. URL `https://www.nora.ai/norsk/news/2022/meet-the-winners-of-fishai.html`.

Jonas Dammen, Åsmund Brekke, Kristian Andersen Hole, Ludvig Løddesøl, Tomas Roaldsnes, and Julia Ortheden. Fishai: The lodestar fishing platform. *Nordic Machine Intelligence*, 2(2):10–12, November 2022. ISSN 2703-9196. doi:10.5617/nmi.9931. URL `https://journals.uio.no/NMI/article/view/9931`.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL `https://arxiv.org/abs/2010.11929`.

Jesús García Fernández, Ismail Alaoui Abdellaoui, and Siamak Mehrkanoon. Deep coastal sea elements forecasting using unet-based models. *Knowledge-Based Systems*, 252:109445, 2022. ISSN 0950-7051. doi:https://doi.org/10.1016/j.knosys.2022.109445. URL `https://www.sciencedirect.com/science/article/pii/S0950705122007110`.

N. Ivošević DeNardis, N. Novosel Vlašić, T. Mišić Radić, J. Zemła, M. Lekka, I. Demir-Yilmaz, C. Formosa-Dague, M. Levak Zorinc, I. Vrana, K. Juraić, L. Horvat, P. Žutinić, M. Gligora Udovič, and B. Gašparović. Behavior and surface properties of microalgae indicate environmental changes. *Journal of Applied Phycology*, 36(1):113–128, February 2024. ISSN 0921-8971, 1573-5176. doi:10.1007/s10811-023-03105-w. URL `https://link.springer.com/10.1007/s10811-023-03105-w`.

Muhammad Kabir, Um E Habiba, Wali Khan, Amin Shah, Sarvat Rahim, Patricio R. De los Rios-Escalante, Zia-Ur-Rehman Farooqi, Liaqat Ali, and Muhammad Shafiq. Climate change due to increasing concentration of carbon dioxide and its impacts on environment in 21st century; a mini review. *Journal of King Saud University - Science*, 35(5):102693, 2023. ISSN 1018-3647. doi:https://doi.org/10.1016/j.jksus.2023.102693. URL `https://www.sciencedirect.com/science/article/pii/S1018364723001556`.

Yuki Kanamori, Toshikazu Yano, Hiroshi Okamura, and Yuta Yagi. Spatio-temporal model and machine learning method reveal patterns and processes of migration under climate change. *Journal of Biogeography*, 51 (4):522–532, April 2024. ISSN 0305-0270, 1365-2699. doi:10.1111/jbi.14595. URL https://onlinelibrary.wiley.com/doi/10.1111/jbi.14595.

Yasmin M. Kassim, O. V. Glinskii, V. V. Glinsky, V. H. Huxley, G. Guidoboni, and K. Palaniappan. Deep u-net regression and hand-crafted feature fusion for accurate blood vessel segmentation. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1445–1449, 2019. doi:10.1109/ICIP.2019.8803084.

Rajaa Kholssi, Hanane Lougraimzi, and Ignacio Moreno-Garrido. Influence of salinity and temperature on the growth, productivity, photosynthetic activity and intracellular ros of two marine microalgae and cyanobacteria. *Marine Environmental Research*, 186:105932, 2023. ISSN 0141-1136. doi:https://doi.org/10.1016/j.marenvres.2023.105932. URL https://www.sciencedirect.com/science/article/pii/S0141113623000600.

Jason Kugelman, Joseph Allman, Scott A. Read, Stephen J. Vincent, Janelle Tong, Michael Kalloniatis, Fred K. Chen, Michael J. Collins, and David Alonso-Caneiro. A comparison of deep learning u-net architectures for posterior segment oct retinal layer segmentation. *Scientific Reports*, 12(1):14888, September 2022. ISSN 2045-2322. doi:10.1038/s41598-022-18646-2. URL https://www.nature.com/articles/s41598-022-18646-2.

Aldrin Lambon, Erica Faye Sagun, Myra Saet, Zyrah Maranon, and Sharmen Berlin. Fishmaze: Fish monitoring and ai-based zone evaluation. *Nordic Machine Intelligence*, 2(2), November 2022. ISSN 2703-9196. doi:10.5617/nmi.9941. URL https://journals.uio.no/NMI/article/view/9941.

Visa Linkiö, Kristian Lahtinen, and Juho Kolmonen. Clusters and traveling fisherman. *Nordic Machine Intelligence*, 2(2): 13–15, November 2022. ISSN 2703-9196. doi:10.5617/nmi.9930. URL https://journals.uio.no/NMI/article/view/9930.

Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention u-net: Learning where to look for the pancreas. In *Conference on Medical Imaging with Deep Learning*, 2018. doi:10.48550/ARXIV.1804.03999. URL https://arxiv.org/abs/1804.03999.

Alba Ordoñez, Ingrid Utseth, Olav Brautaset, Rolf Korneliussen, and Nils Olav Handegard. Evaluation of echosounder data preparation strategies for modern machine learning models. *Fisheries Research*, 254:106411, 2022. ISSN 0165-7836. doi:https://doi.org/10.1016/j.fishres.2022.106411. URL https://www.sciencedirect.com/science/article/pii/S0165783622001886.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Remote Sensing Systems. RSS SMAP Level 3 Sea Surface Salinity Standard Mapped Image Monthly V4.0 Validated Dataset, 2019. URL https://podaac.jpl.nasa.gov/dataset/SMAP_RSS_L3_SSS_SMI_MONTHLY_V4.

Richard W. Reynolds, Thomas M. Smith, Chunying Liu, Dudley B. Chelton, Kenneth S. Casey, and Michael G. Schlax. Daily High-Resolution-Blended Analyses for Sea Surface Temperature. *Journal of Climate*, 20(22): 5473–5496, November 2007. ISSN 1520-0442, 0894-8755. doi:10.1175/2007JCLI1824.1. URL http://journals.ametsoc.org/doi/10.1175/2007JCLI1824.1.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. URL http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a. (available on arXiv:1505.04597 [cs.CV]).

Tor-Arne Schmidt Nordmo, Ove Kvalsvik, Svein Ove Kvalsund, Birte Hansen, and Michael A. Riegler. Fish AI: Sustainable Commercial Fishing. *Nordic Machine Intelligence*, 2(2):1–3, June 2022. ISSN 2703-9196. doi:10.5617/nmi.9657. URL https://journals.uio.no/NMI/article/view/9657.

Najmeh Sheikhzadeh, Mehdi Soltani, Marzieh Heidarieh, and Mahdi Ghorbani. Role of dietary microalgae on fish health and fillet quality: Recent insights and future prospects.

Fishes, 9(1):26, January 2024. ISSN 2410-3888. doi:10.3390/fishes9010026. URL https://www.mdpi.com/2410-3888/9/1/26.

U. Rashid Sumaila and Travis C. Tai. End overfishing and increase the resilience of the ocean to climate change. *Frontiers in Marine Science*, 7, 2020. ISSN 2296-7745. doi:10.3389/fmars.2020.00523. URL https://www.frontiersin.org/journals/marine-science/articles/10.3389/fmars.2020.00523.

Craig Syms. Satellite ocean data can inform precision fishing. *Nordic Machine Intelligence*, 2(2), November 2022. ISSN 2703-9196. doi:10.5617/nmi.9945. URL https://journals.uio.no/NMI/article/view/9945.

Kevin Trebing, Tomasz Stańczyk, and Siamak Mehrkanoon. Smaat-unet: Precipitation nowcasting using a small attention-unet architecture. *Pattern Recognition Letters*, 145:178–186, 2021. ISSN 0167-8655. doi:https://doi.org/10.1016/j.patrec.2021.01.036. URL https://www.sciencedirect.com/science/article/pii/S0167865521000556.

Armando J. Ubeda, C. A. Simpfendorfer, and M. R. Heupel. Movements of bonnetheads, sphyrna tiburo, as a response to salinity change in a florida estuary. *Environmental Biology of Fishes*, 84(3):293–303, March 2009. ISSN 0378-1909, 1573-5133. doi:10.1007/s10641-008-9436-5. URL http://link.springer.com/10.1007/s10641-008-9436-5.

Department of Economic United Nations and Social Affairs Sustainable Development. Transforming our world: the 2030 agenda for sustainable development, 2015. URL https://sdgs.un.org/2030agenda.

Yimin Yang and Siamak Mehrkanoon. Aa-transunet: Attention augmented transunet for nowcasting tasks, 2022. URL https://arxiv.org/abs/2202.04996.

Wei Yao, Zhigang Zeng, Cheng Lian, and Huiming Tang. Pixel-wise regression using u-net and its application on pansharpening. *Neurocomputing*, 312:364–371, 2018. ISSN 0925-2312. doi:https://doi.org/10.1016/j.neucom.2018.05.103. URL https://www.sciencedirect.com/science/article/pii/S0925231218307008.

Xi Ye and Guillaume-Alexandre Bilodeau. Vptr: Efficient transformers for video prediction, 2022. URL https://arxiv.org/abs/2203.15836.

Lirong Yin, Lei Wang, Tingqiao Li, Siyu Lu, Jiawei Tian, Zhengtong Yin, Xiaolu Li, and Wenfeng Zheng. U-net-lstm: Time series-enhanced lake boundary prediction model. *Land*, 12(10), 2023. ISSN 2073-445X. doi:10.3390/land12101859. URL https://www.mdpi.com/2073-445X/12/10/1859.

A Appendix

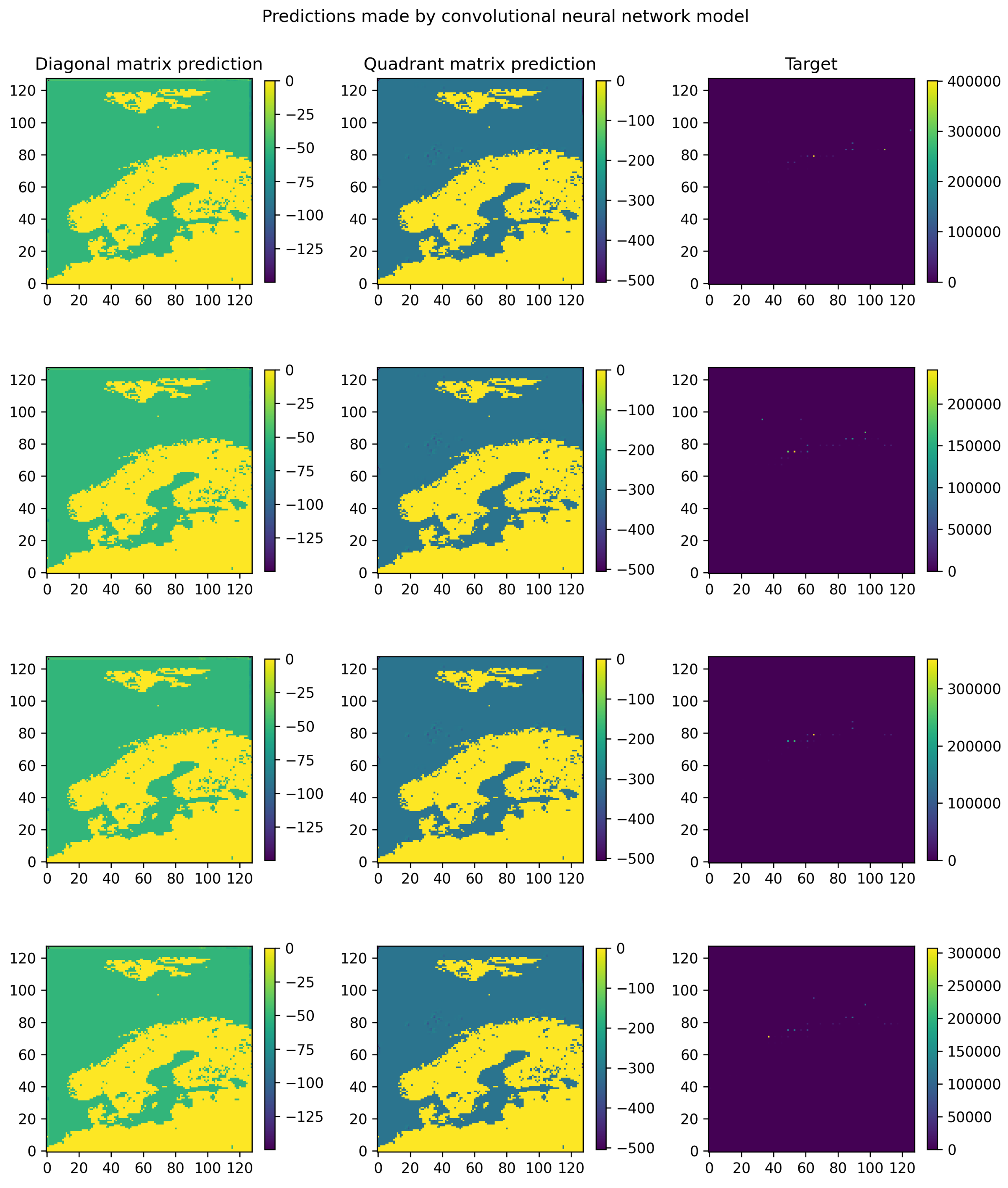

Figure A.1: Four predictions made by the standard convolutional networks.

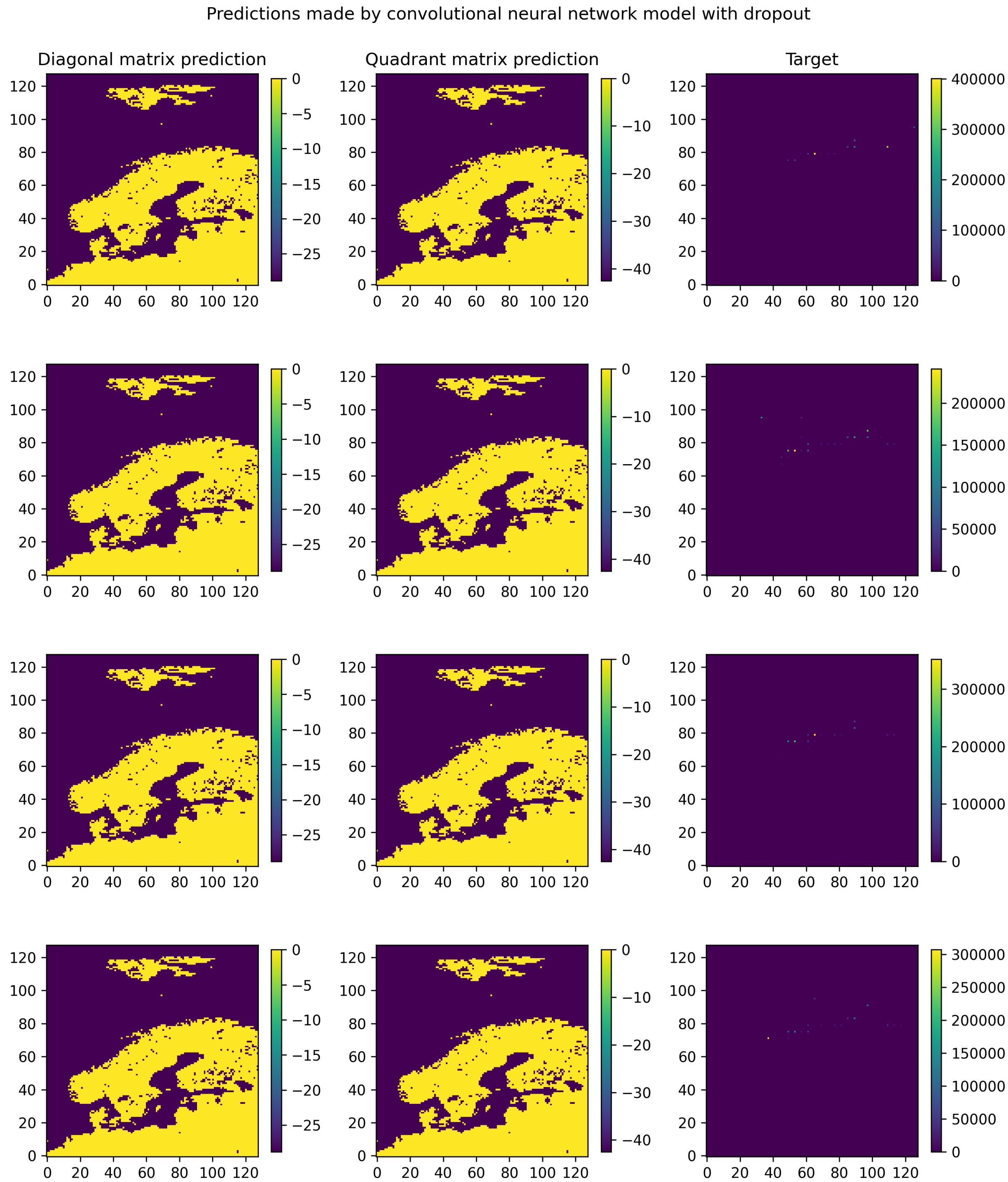

Figure A.2: Four predictions made by the convolutional networks with dropout.

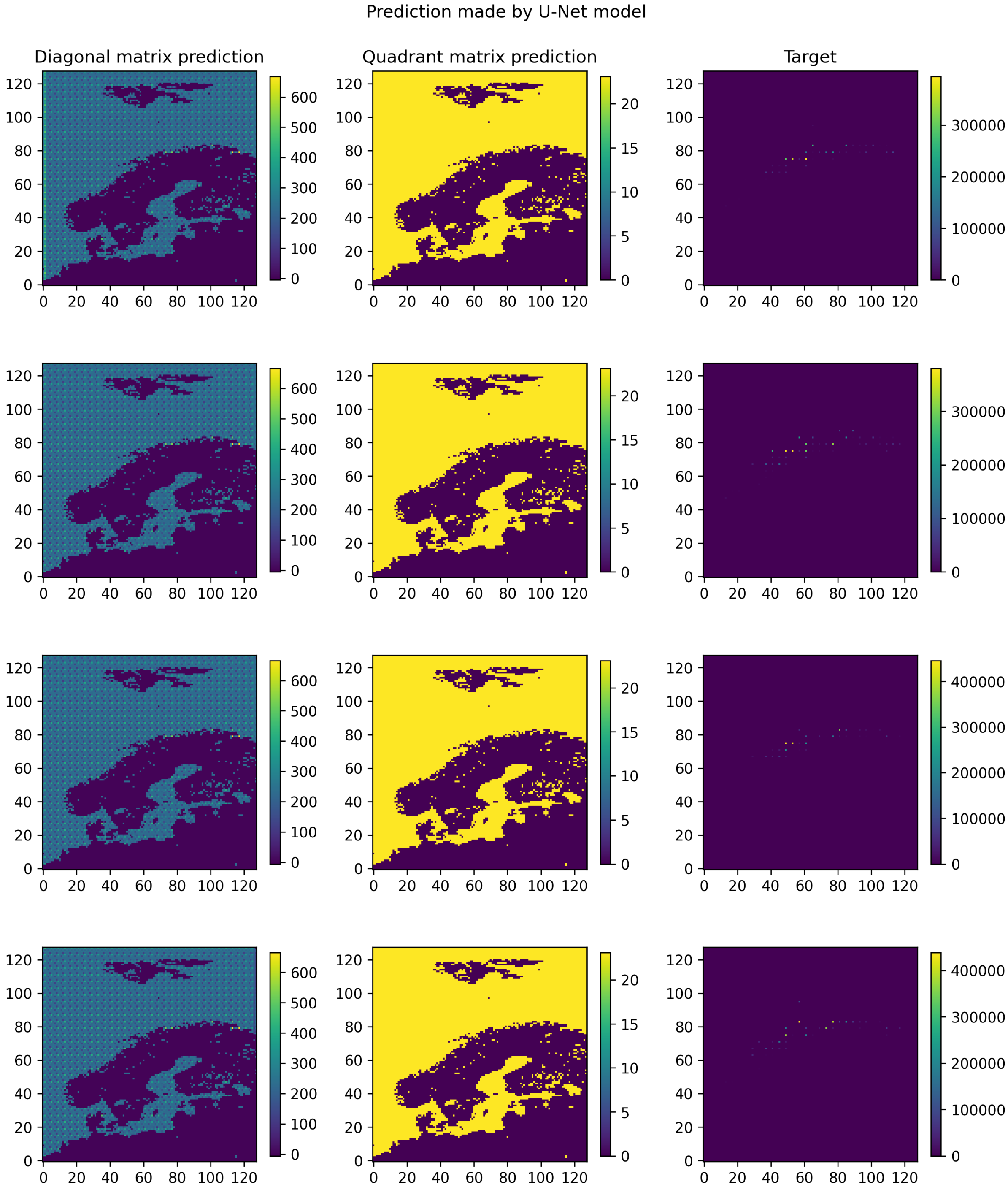

Figure A.3: Four predictions made by the standard regression U-Net model.

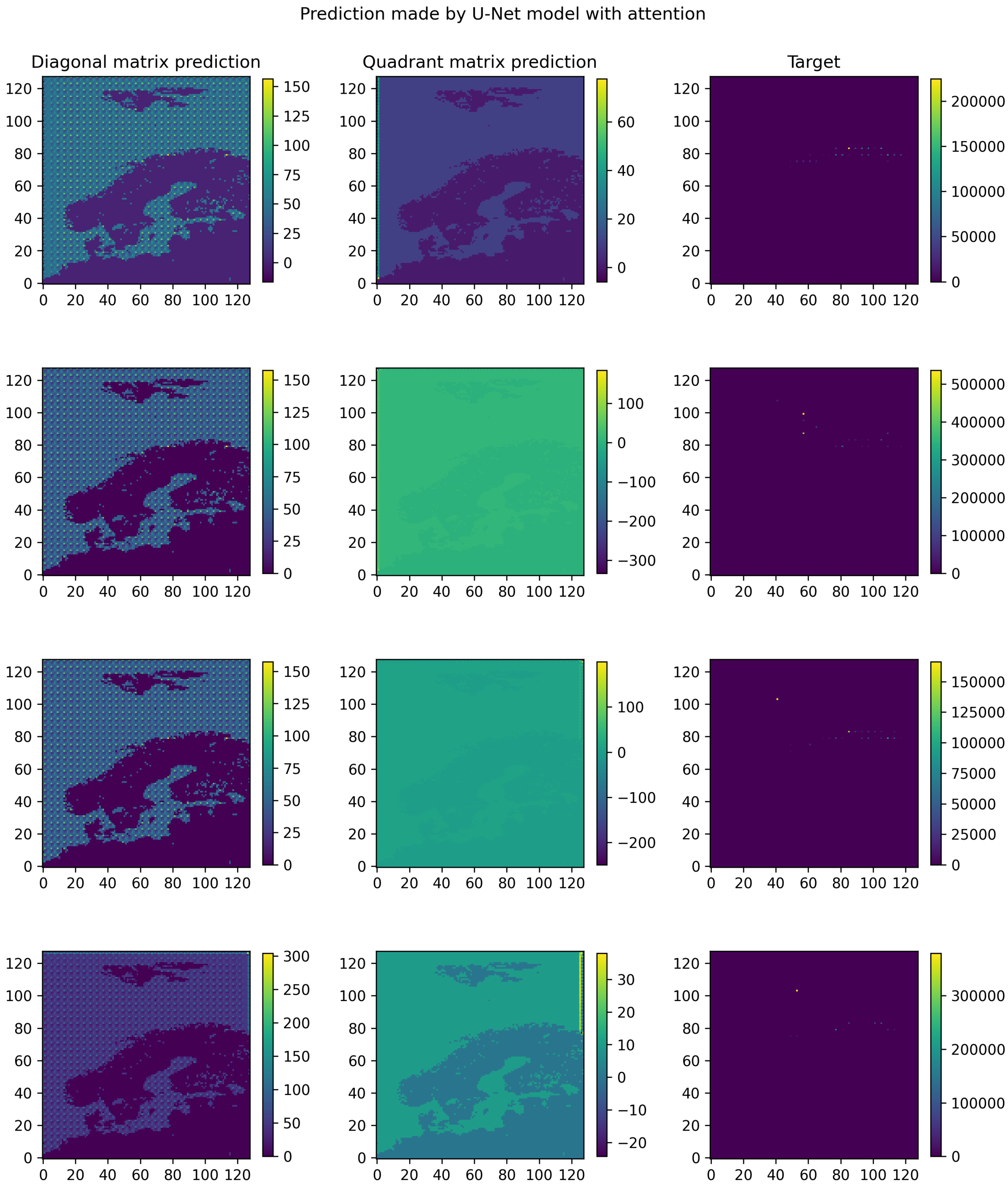

Figure A.4: Four predictions made by the regression U-Net model with attention gates.

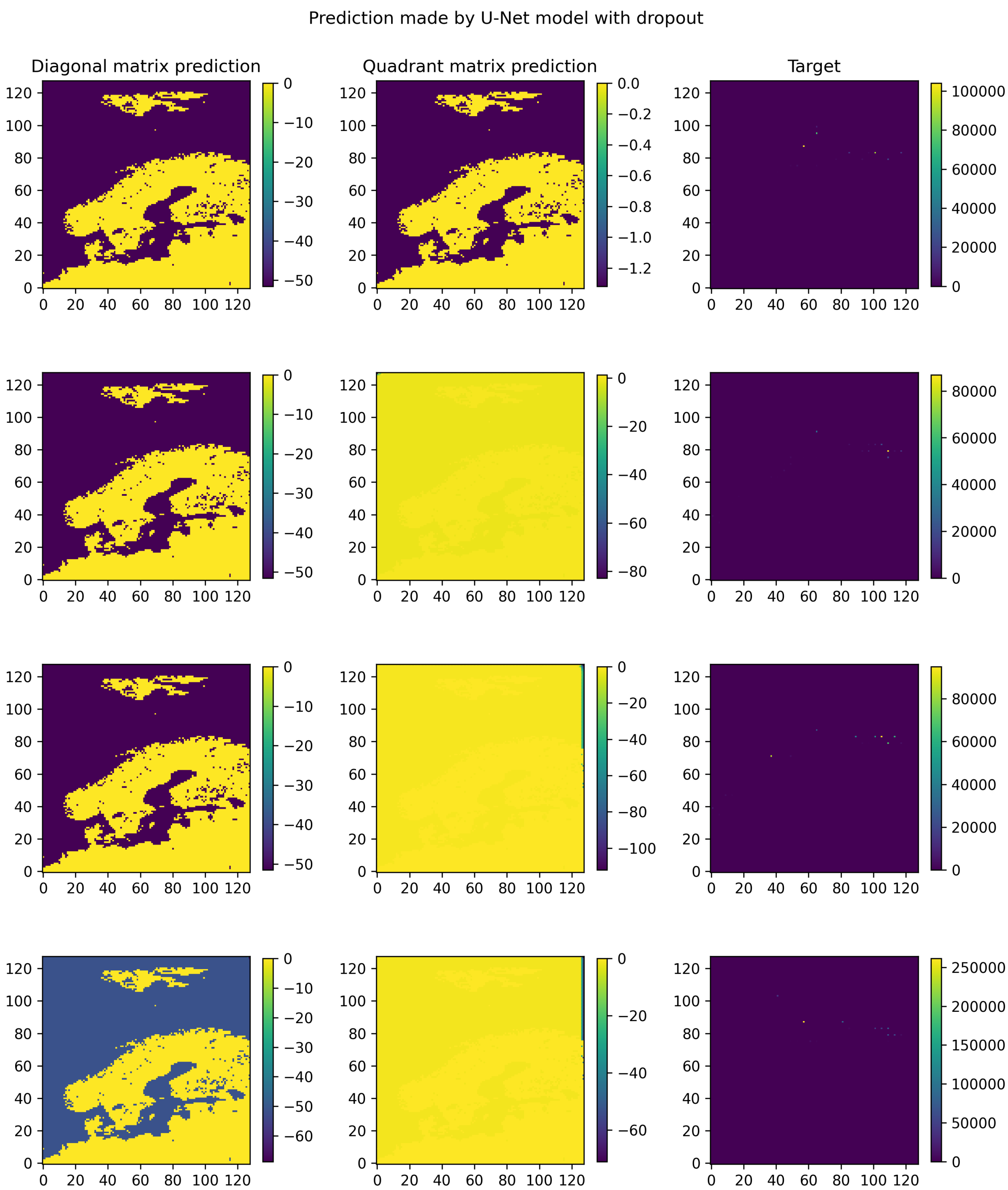

Figure A.5: Four predictions made by the regression U-Net model with dropout.

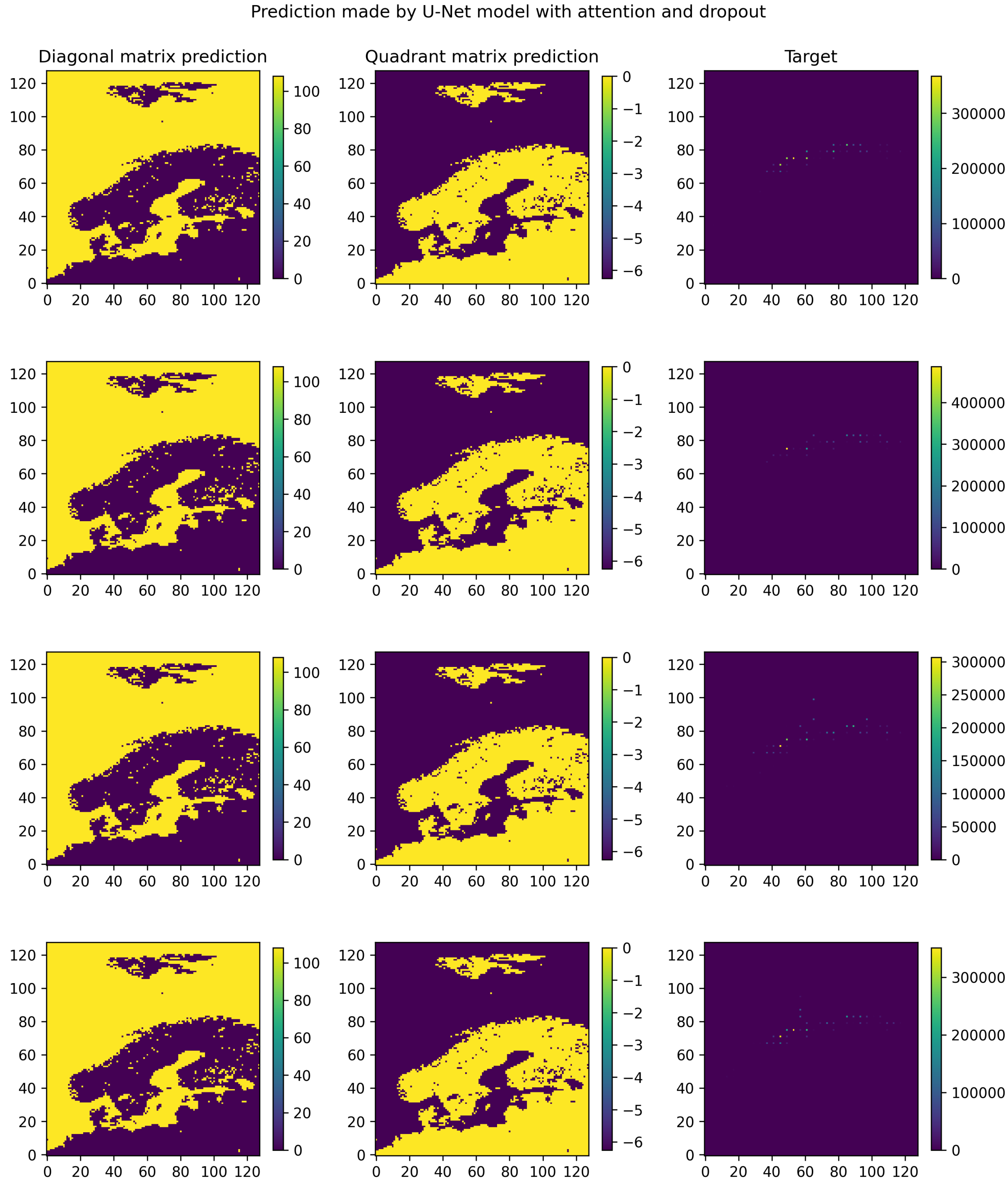

Figure A.6: Four predictions made by the regression U-Net model with attention gates and dropout.

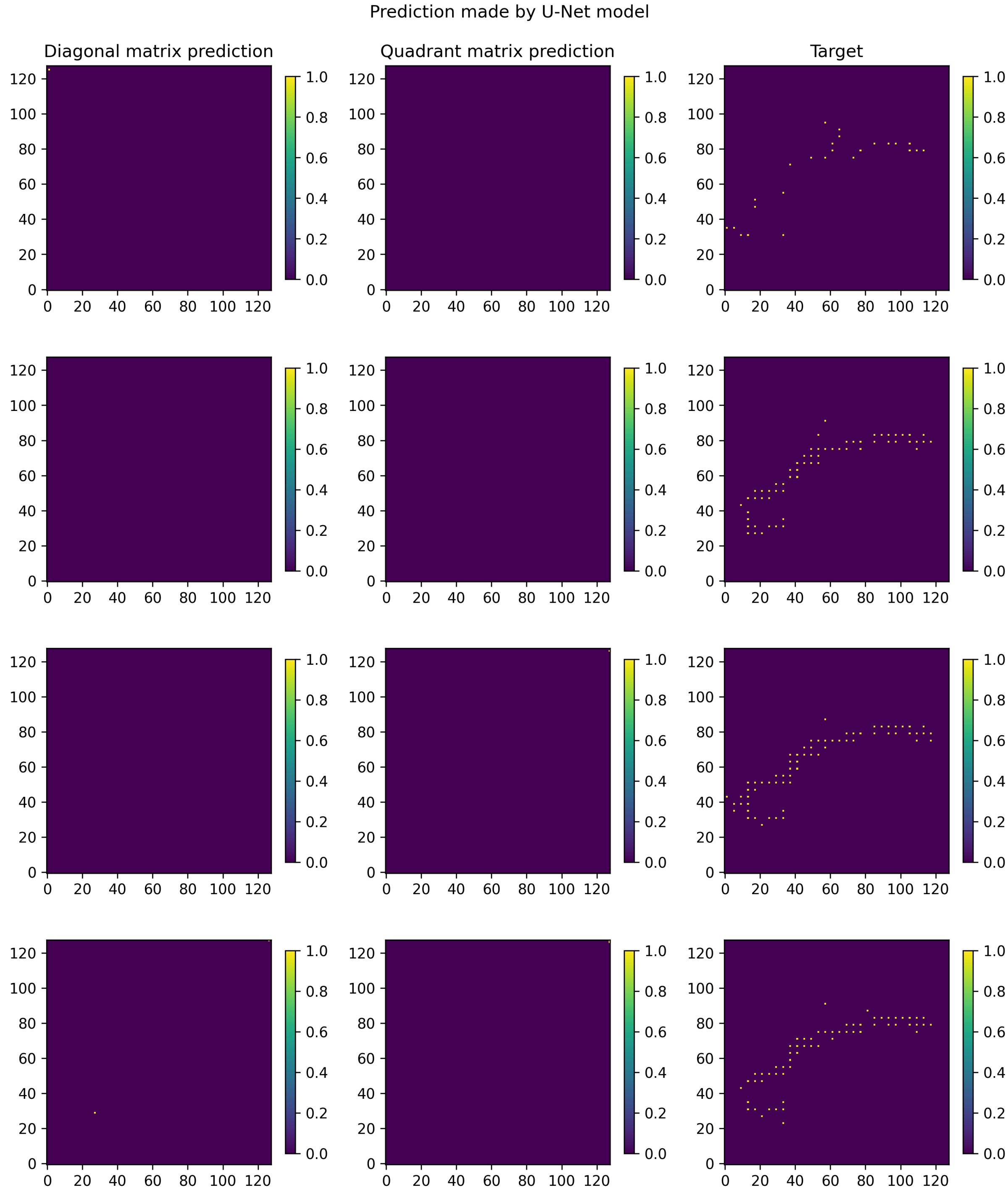

Figure A.7: Four predictions made by the standard classification U-Net model.

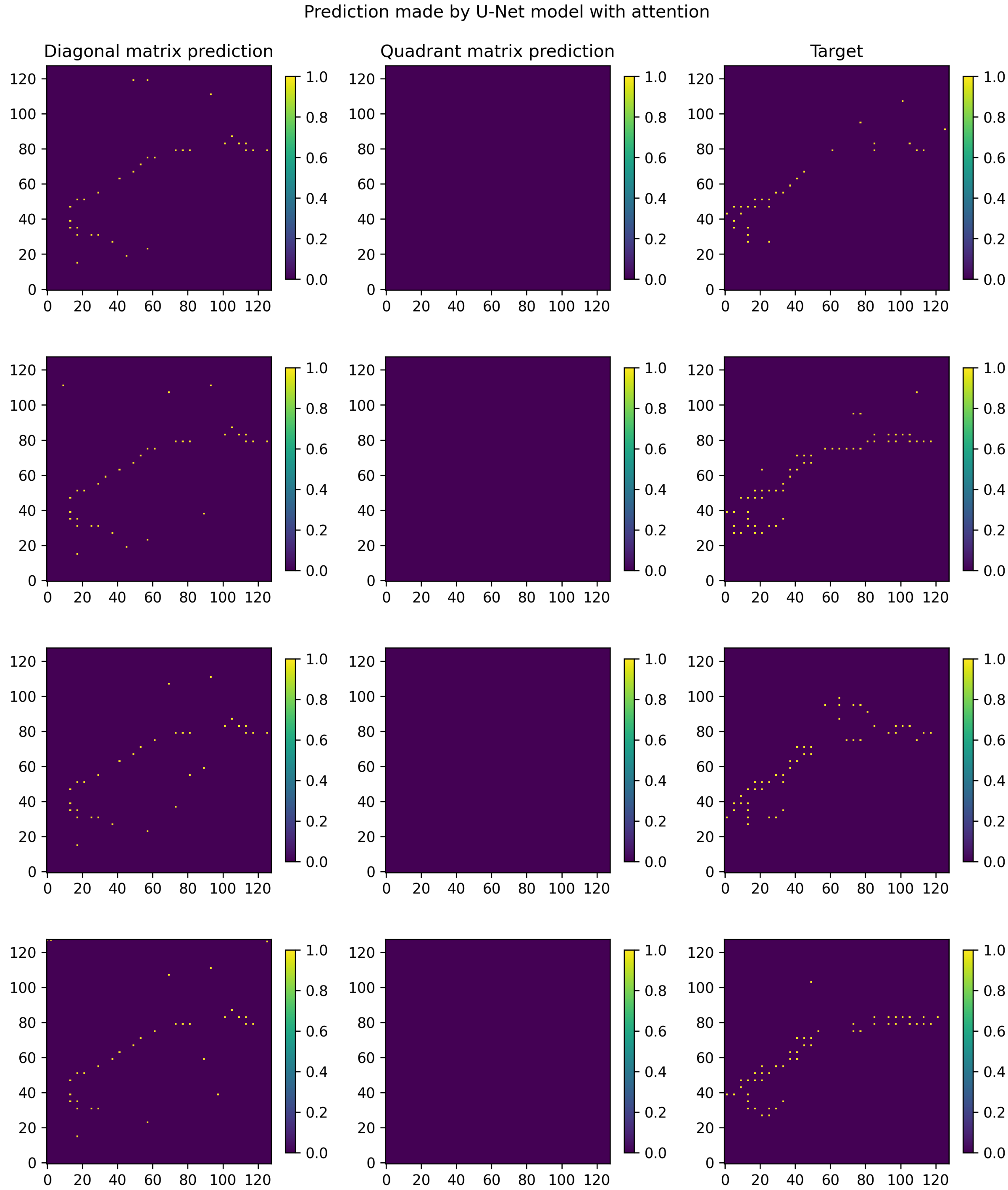

Figure A.8: Four predictions made by the classification U-Net model with attention gates.

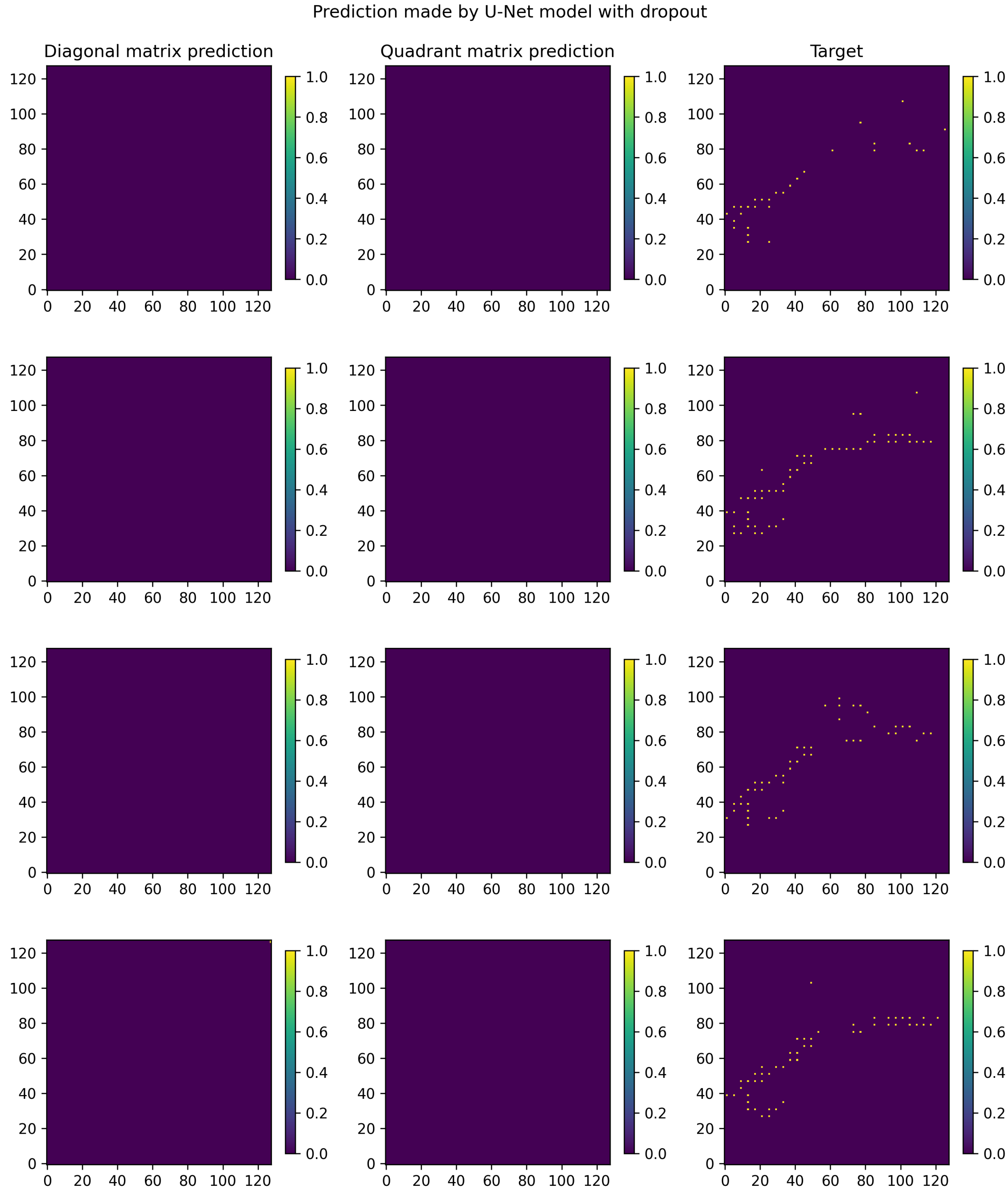

Figure A.9: Four predictions made by the classification U-Net model with dropout.

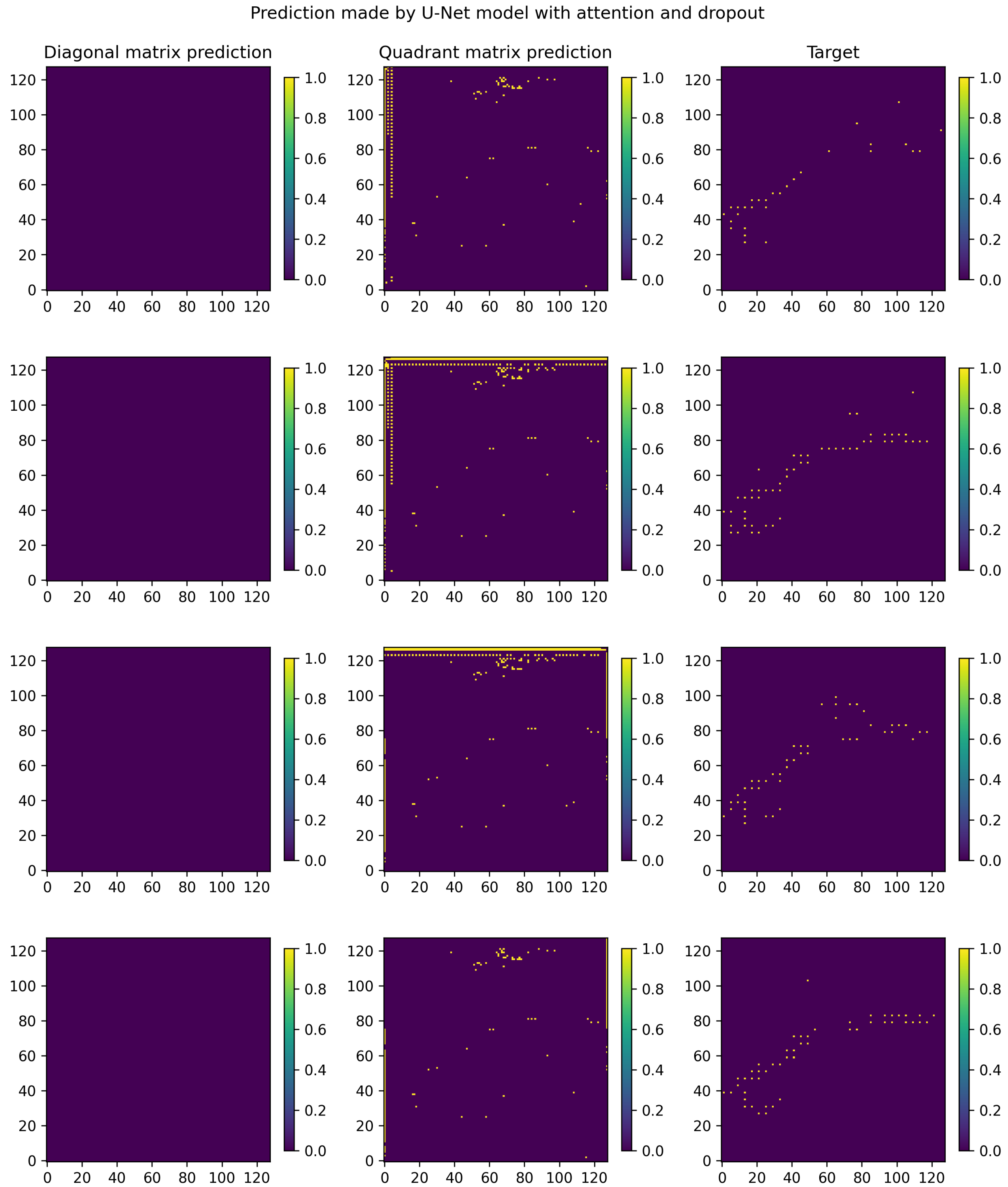

Figure A.10: Four predictions made by the classification U-Net model with attention gates and dropout.