UNIVERSITY OF GRONINGEN

Bachelor Thesis Report

Student:
Cristian Mihai ROSIU - s3742377

Supervisors:
Estefanía TALAVERA MARTÍNEZ
Dimka KARASTOYANOVA

July 15, 2021

Multi-task learning using deep networks for the classification of seals auscultations

Cristian Mihai Rosiu

***Abstract*—Pulmonary auscultation is one of the most valuable and fundamental tools available to veterinarians to quickly assess lung conditions in animal. Despite recent advances in the medical field, electronic chest auscultation is still considered a less reliable method, mainly due to the non-stationary property of lung sounds. Automating this process can aid the clinicians in their diagnosis process and hopefully improve the survival rate of seals that arrive at the Pieterburen Zeehondencentrum in Groningen. One way to do this is through means of deep learning. However, in most papers, audio classification tasks are usually treated as independent tasks. As lung sounds are known to be related to one another, a single-task approach misses most of the information necessary to make a difference when classifying closely related tasks. This paper aims to show the potential of multi-task learning (MTL) in the context of seal lung sounds classification. We proposed two different types of multi-task convolutional neural network architectures. These models are evaluated on the mel-cepstral coefficients (MFCCs) features and per-channel energy normalized spectrograms (PCEN). Experiments were conducted on a dataset of 142 samples gathered from both the left and right lungs of seals that arrive at the sanctuary. The two types of abnormal sounds present in this dataset are Wheezing and Rhonchus. Results show that the MFCC features, together with our custom-built CNN obtained an accuracy of 73% when classifying wheezing and 63% in the case of rhonchus, outperforming the classification of PCEN images by 15% and 25% respectively. Lastly, the same model manage to obtain a survival prediction accuracy of 80% and successfully showing the potential of MTL in auscultation classification.**

***Index Terms*—Multi-task learning Convolutional Neural Networks Deep Residual Networks Lung Audio Classification**

I. Introduction

LUNG sounds characteristics and diagnoses form an indispensable part of pulmonary pathology. These sounds convey essential information about the state of an animal's respiratory systems. Veterinarians can use diverse methods to record and identify sounds and respiratory problems, but most of the available methods are not always convenient nor reliable [1]. Electronic chest auscultation is a cheap, save, non-invasive and easy to perform technique [2][3] that helps clinicians in their diagnosis of pulmonary diseases [4][3]. Furthermore, pulmonary auscultation is one of the most useful and fundamental tools available to veterinarians to quickly assess lung condition in animals [2]. In this method, the medic uses the stethoscope to hear normal, decreased or absent, and abnormal breath sounds. However, the interpretation of the sounds heard during auscultation and its use as a diagnostic technique are directly related to the experience of the performer, hence defining this technique as subjective [2][4][5][6]. Another important factor that makes auscultation a complex and difficult process is the human ear, which is usually sensitive to either very high or shallow frequency bands and might lead to the inability of hearing abnormal sounds. Furthermore, the non-stationary nature of a lung sound makes the detection task even more difficult for the veterinarians [7].

Deep learning methods can help veterinarians automate the process of diagnosing abnormal sounds found in seal lungs in order to increase their survival rate. Despite the benefits brought by the advances in machine learning, most of the lung audio classification techniques such as emotion recognition [8], accent classification [9] and even natural language processing (NLP) [10] treat classification tasks separately. Because some of these tasks are closely related, a single-task approach might lose important information needed to help do better on the optimization of a certain metric. For example, while abnormal sounds like Rhonchus and Wheezing can be regarded as separate classification tasks, in most cases, these sounds can influence the survival rate of a seal, thus making them closely related.

One way to exploit standard features between related tasks is through means of Multi-task Learning (MTL). Generally, when more than one metric aims to be optimized, we are effectively doing multi-task learning [11]. This method can really improve the generalization capabilities of a deep model by training all tasks simultaneously using the same shared environment. This shared environment introduces something called an inductive bias which is mainly created by auxiliary tasks and makes the model prefer hypotheses that are able to explain more than one task. In essence, this means that the architecture will strive to adjust its weights such that it satisfies all the classification tasks present in the input space, instead of just optimizing for only one task [11].

Borrowing this idea, we have built two different models using an MTL architectural design. The aim of this paper is to present our attempt at using MTL as a tool that can help exploit the relationships between the Wheezing, Rhonchus and survival classification tasks by jointly learning from them. The main two inputs to our models are the mel-cepstral coefficients (MFCC) and mel-spectrograms, to which we apply per-channel energy normalization. These two ways of visualizing audio signals are the backbone of audio classification. The MFCC feature extraction technique is one of the most popular spectral based parameters used in the recognition system approach [1]. On the flipside, spectrograms are detailed image representation of audio signals that are also widely used in audio classification tasks [12][13][14].

The rest of the paper is organized as follows: In Section 2 we talk about some related works in the field of multi-task learning and lung sound classification. Next, in Section 3, we present the methods and tools used in this paper such as pre-processing, feature extraction techniques and types of MTL

architectures. Section 4 talks about the experimental setup including information about dataset, validation techniques used and implementation details. Obtained results and the discussion are present in Section 5. Lastly, Section 6 contains the conclusion and feature works of this paper.

II. Related Work

Automatic lung sound classification has received substantial attention from the research community in recent years. Because of their non-invasive, more economical, and patient-friendly nature, computerized methods have become more desirable. Signal processing and deep learning methods have proven themselves to be strong candidates when applied to automating human lung sound classification. Most of these methods are adapted for the single audio classification task and have furnished new insights into the analysis of lung sounds for diagnostic purposes.

In the early stages of lung sound classification, research was mostly based on standard machine learning techniques such as self-organizing maps [15], Gaussian mixture models (GMM) [16] , and Support Vector Machines (SVM) [17]. SVMs are potent tools that proved their potential before. In [18], Semra İçer used a support vector machine (SVM) to distinguish between the crackle, Rhonchus, and normal lung sounds, making use of power spectral density (PSD) and instantaneous frequency (IF) features. The author obtained an astonishing accuracy between 90% and 100%, successfully showing that the selected method correctly represented the characteristic changes in sound. Although these standard machine learning algorithms can do relatively well on small data sets with few outliers, their limitations become apparent when used in more complex problems such as image classification and speech recognition.

On the other hand, deep learning techniques became one of the main approaches for adventitious sound detection and classification. These algorithms are powerful processing technologies, which have shown to be very capable in the field of sound analysis and computer vision. Especially, deep neural network classifiers have proven their potential of capturing energy modulation patterns across time and frequency when applied to spectrogram-like inputs. Classifiers such as Convolutional Neural Networks (CNN) [19], Recurrent Neural Networks (RNN) or a combination of CNNs and RNNs have proven to be the most successful approaches [17]. Dalal Bardou et al., released a paper [19] demonstrating how CNNs can easily outperform methods such as K-Nearest Neighbours (KNN), GMM, and SVM. The authors managed to build and train a Convolutional Neural Network on MFFCs features which reached an accuracy level of 93.26% and 95.10% when augmentation techniques were used. Alongside Deep Neural Networks (DNN), the MFCC feature extraction technique has been a de facto standard for speech recognition tasks. However, recent research has led to discoveries of more new and enhanced unsupervised audio feature extraction methods. Nevertheless, One of the most noticeable limitations of deep learning methods is their necessity for immense data-sets which need hand labelling.

All the methods above focus on the single audio classification task. There is little to no study about the potential of the multi-task learning (MTL) methods when applied to the field of lung audio classification. Multi-task learning is a sub-field of machine learning in which multiple tasks are performed in parallel. MTL methods have the potential of exploiting intrinsic relationships among related tasks. Specifically, in [20] a multi-task model for audio classification was designed to exploit the common relations found in audio and to deal with jointly classification. The main goal of MTL is to improve generalization performance by shared learning between DNN, allowing separate networks to learn from one another [20]. To my knowledge, this research is the first one to try and apply technologies such as MTL and DNN to the field of seals lung sound classification.

III. Methodology

This section describes how the data was pre-processed for training and the Convolutional Neural Network (CNN) architectures that we have built.

A. Data Pre-processing

The main three properties of an audio file are: (a) audio channels, (b) sampling rate and (c) bit depth. These properties usually need pre-processing to ensure consistency across the whole data set. Given that the same device recorded the audio, the properties are already shared by the audio samples. Usually, the audio samples are pre-processed following a standard procedure, firstly, they are normalized by forcing the bit-depth values to range between -1 and 1 and flattening the audio channels into mono. Later, zeros are added, in a process named zero-padding, for samples that had less than 30 seconds such that each audio feature has the same duration when later processed.

B. Feature Extraction

In order to train our model, we have to extract the right features. There are a handful of feature extraction methods; however, there is no standard technique that can help maximize the accuracy of a model. How good a method performs, is purely given by the type of problem we are trying to solve. The following feature extraction techniques were used in this paper:

1) MFCCs: The use of the MFCCs feature extraction technique has been a de facto standard for speech recognition tasks for a long time. This method is one of the most popular spectral based approaches used in recognition systems. These coefficients are a small set of features (usually between 10-20) which summarises the frequency distribution across the window size. They are derived from a type of cepstral representation of the audio clip. The use of this method was motivated by their success in the fields of automated sound classification, and speech recognition [1]. In speech processing, MFCCs are used to represent the spectral envelope that separates the source from the filter. Because of this trait, these coefficients have also been proven efficient when applied

to the field of respiratory sounds [1]. In the context of lung audio signals, the thoracic wall acts as the filter through which the lung sounds sift [1]. The features can also approximate the human auditory system's response, allowing for a better representation of sound. These coefficients are derived as follows:

1) Take the Fourier Transform of a windowed signal (2), where $g(y)$ representa a signal with zero initial phase and frequency f_0 and $e^{-2\pi iyx}$ a sinusoidal signal

$$f(x) = \int_{-\infty}^{\infty} g(y) e^{-2\pi iyx}, dx \tag{1}$$

2) Map the obtained spectrum onto the mel scale.
3) Take the logs of the powers at each of the mel frequencies
4) Take the discrete cosine transform of the mel log powers form the previous step
5) The resulting amplitudes represent the MFCC features

A visual representation of these features can be seen in Figure 1. The images show the features extracted from four different audio clips. Lastly, each feature has assigned a color depending on how frequent it is in the cepstral representation.

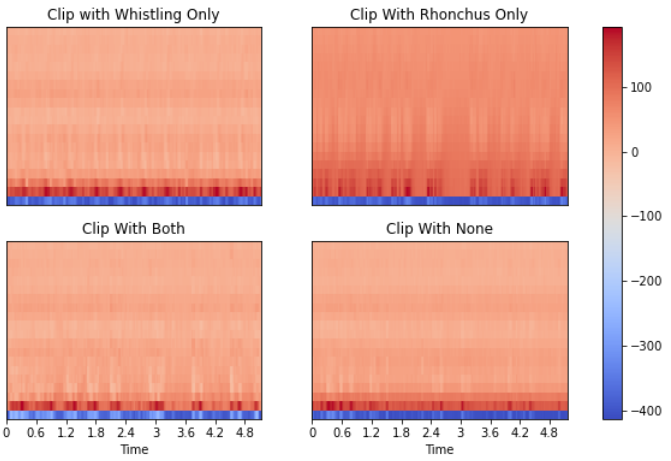

Fig. 1: All images show the visual representation of the extracted mel-cepstral coefficients for four different audio files: no sounds, only Wheezing, only Rhonchus and both sounds.

2) PCEN Mel Spectrograms: Adjusting the perceptual mel-scale provides a time-frequency image representation, named mel-frequency spectrogram. However, the noise made by tiny dust particles or the movement of the stethoscope is usually detrimental to the equivariance along the mel-frequency axis [21][22]. Therefore, we used a relatively new technique of logarithmic transformation called per-channel normalization (PCEN) in order to reduce the channel distortions in our audio files [23]. PCEN combines dynamic range compression (DRC) and adaptive gain control (AGC) with temporal integration. While AGC is intended to supress stationary background noise, DRC reduces the variance of the foreground loudness [23]. Formula (2) shows how PCEN is computed, where $E(t, f)$ are the magnitudes in a mel-frequency spectrogram of real-world acoustic scenes and they are correlated, both along time t and mel frequency f. Lastly, α, ϵ, r and δ are just positive constants.

A visual example of the capabilities of this technique can be seen in Figure 2 where it was applied to a boisterous image that contained Wheezing and Rhonchus.

$$\textbf{PCEN}(t, f) = \left(\frac{E(t, f)}{(\epsilon + (E^t * \theta_r)(t, f))^\alpha} + \delta\right)^r - \delta^r \tag{2}$$

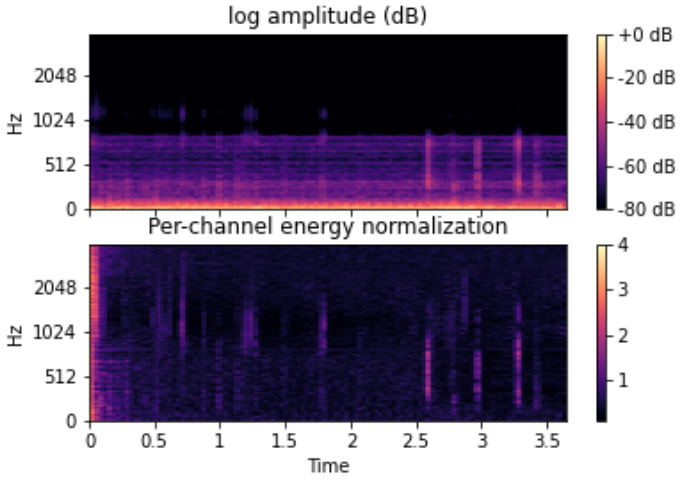

Fig. 2: The top image represents the mel-spectrogram of a noisy audio containing only Rhonchus. The bottom images shows the same spectrogram after PCEN was applied. It also shows how most of the noise is removed, only leaving the most important features of the mel-spectrogrma.

C. Multi-Task Learning (MTL)

There are two most common methods of applying MTL in a deep learning context. The first approach is to use Soft Parameter Sharing where each task has its own model and parameters [11]. However, the distance between all parameters is usually regularized using the L2 norm to impose a degree of similarity between all of the shared parameters [24].

The second method is called Hard Parameter Sharing and its architecture can be observed in Figure 3. It is one of the most commonly used MTL approaches in Deep Learning [11]. The idea behind it is that the hidden layers are shared between all tasks while having some separate task specific output layers.

As our data-set significantly lacks more samples, over-fitting is inevitable. Therefore, we have chosen to follow the Hard Parameter sharing architecture as it has been shown that it can heavily reduce the risk of over-fitting. In multi-task learning, the risk of over-fitting the shared parameters is an order of N (i.e. the number of tasks) [25]. Intuitively, the more tasks are there for the model to learn, the greater the chance of better generalization. This is natural as the model now has to find a better representation that encapsulates all classification tasks present in our input space.

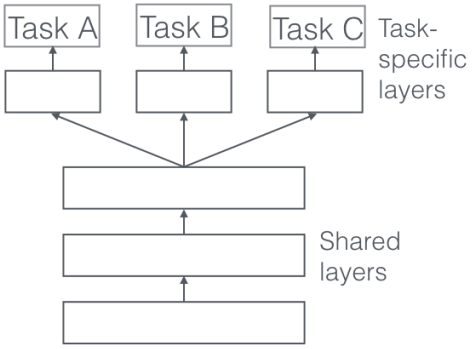

Fig. 3: Multi-task Hard Parameter Sharing architecture diagram. It can be seen that the bottom layers share the weights of the tasks. Then, the network branches out in N separate task-specific output layers - where N is the number of classification tasks-

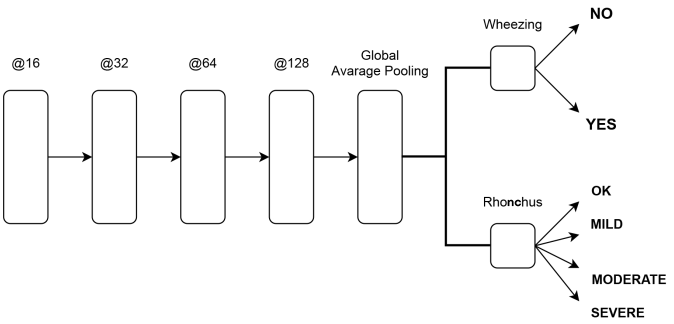

Fig. 4: Our custom built CNN model. It contains four shared convolutional blocks. These blocks are then connected to a global average pooling (GAP) layer. Finally, the model branches out into two separate classification branches, one for the Wheezing and one for the Rhonchus.

Multi-task learning also comes with some important characteristics which makes it such a powerful tool. The first one is implicit data augmentation. MTL increases the samples size used for training our model. When a model is trained on a task, ideally, the end goal would be to make it ignore data-dependent noise and generalize well. If a model jointly learns from multiples task, it enables it to obtain a better representation thorough averaging the noise patterns [11].

Attention focusing is another MTL property. When the data is noisy or limited, the model struggles to differentiate between relevant and irrelevant features. MTL can help the model to switch its attention on features that actually are important as other tasks will provide bias with regards to the relevance or irrelevance of a certain feature, and thus, making it easier for a model to chose a general representation [11].

D. Multi-Task Convolutional Neural Network

We have built two different types of multi-task CNNs models. These models are built around the hard parameter sharing architecture. This means that all the hidden layers are shared between all our tasks while keeping some specific output layers separate.

1) Custom CNN: The architecture of our custom build CNN can be seen in Figure 4. It has four shared convolutional layers. After each convolutional layer, a dropout layer is applied. Dropout is a regularization method in which some number of output layers are randomly ignored. This significantly reduces the overfitting induced by the low sample size. The output layer of our convolutional block is feature map with size (1x1x128) which is fed into a global average pooling layer. The idea behind this choice is to replace the fully connected layers by generating one feature map for each corresponding category of the classification task. One of the advantages of this technique is that there are no parameters to optimize. Therefore, overfitting is omitted at this step. Finally, these feature maps are directly fed into either two separate output Dense layers, in the case of our Wheezing and Rhonchus classification, or into three output layers in when the Survival task is added.

2) Residual Networks: Our second approach for a model architecture was to use the convolutional layers blocks of a non-trained ResNet50 model as the shared layers between our tasks. Residual Networks are a type of Artificial Neural Network (ANN), which can mitigate the vanishing gradient problem that profound networks have [26]. This is done through skip connections, which allows gradient information to be passed between layers to prevent minimal values.

In order to implement a MTL approach we had to disabled the fully connected layers and only kept the convolutional blocks. The output of the final conventional block is then fed into a global average pooling which branches out into two specific output Dense layers. A very minimal example of this can be seen in Figure 5.

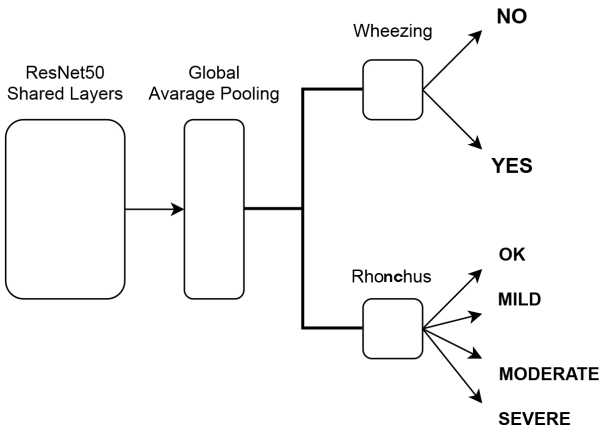

Fig. 5: The model used for PCEN classificaition. The shaed convolutional layers are borrowed from the ResNet50 non-trained model. These layers are then connected to a GAP layer. Finally, the model branches out into the Wheezing and Rhonchus Dense output layers.

The activation function used with our models was softmax (6) in case of Rhonchus and sigmoid (5) for the binary Wheezing and survival tasks. Finally the two loss functions we used were the sparse categorical crossentropy (4) toghether

with the binary crossentropy (3)

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i * log(p(y_i)) + (1-y_i) * log(1-p(y_i)) \quad (3)$$

$$L(\Theta) = -\sum_{i=1}^{K} y_i log(\hat{y}_i) \quad (4)$$

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1} = 1 - S(-x) \quad (5)$$

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad (6)$$

IV. Experimental Framework

A. Dataset

The data-set was gathered by seals veterinarians using a 3M Littmann Electronic Stethoscope. The use of a stethoscope, either conventional or electronic, helps clinicians in their diagnosis of pulmonary diseases. This kind of stethoscope can amplify the sounds and permit the recording of those for future processing and analysis. This data set contains 146 various audio recordings of chest auscultations during intake, sampled at a rate of 4000Hz by the electronic stethoscope. The audio files come from both the left and right lungs of a seal with each lung being assigned one or more diagnosed sounds (i.e. Rhonchus and Wheezes), together with their severity level (i.e. Rhonchus) or existence (i.e. Wheezing)

In Figure 6, we can see the class distributions of each task. Firstly, our Wheezes distribution is the most balanced one, with a difference of only 17% between positive and negative classes. On the other hand, looking at the Rhonchus class distribution, we can clearly observe how the severe and moderate labels make up only 23% out of the total samples. Furthermore, a similar unbalanced pattern can also be observed in the survival pie chart, where only 22% out of the total number of samples have actually died.

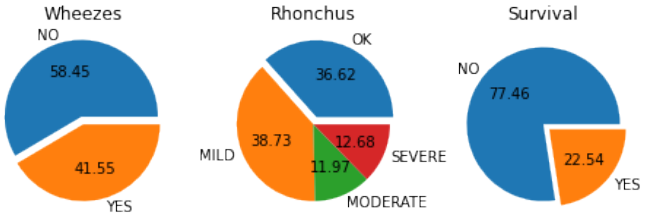

Fig. 6: Three pie-charts showing the distribution of classes for each task. The left plot shows 83 samples belonging to the negative class and only 59 to positive class. On the middle image, from the total of 146 audio files, 53 belong to the a healthy lung sounds, 55 to the mild Rhonchus sounds, and the rest of 34 to the moderate and severe classes. Lastly, in the case of survival, 109 samples are related to the survival of an animal and 31 with its death

B. Validation

To evaluate the proposed CNN architecture, multiple evaluation metric methods were applied in order to make sure that the model outputs the most optimal results.

1) Confusion Matrix: The confusion matrix is a way to visualize how hard the computer struggled to differentiate between classes. In other terms, a measure of how well the chosen CNN architecture has performed on the given data after training. In our confusion matrix, the X-axis represents the predicted labels while the Y-axis represents the actual labels. Blue coloured cells that are situated on the first diagonal of this matrix contain the number of samples that the model accurately predicted. Contrary to that, white cells contain the number of samples that were incorrectly predicted.

Using the confusion matrix we can extract important metrics that help visualize what a model is capable of. In the classification reports found in this paper there are present 3 additional metrics: Precision (4), Recall (5), and F1 (6) scores. Precision is the ratio of correctly predicted positive observations out of the total correctly predicted positive and negative labels. Similarly to precision, recall is the ratio of the correctly predicted positive labels to all the observations in a specific class. Lastly, the F1 score is the weighted average of precision and recall. The F1 score is usually more useful than accuracy, especially when there is an uneven class distribution.

$$\textbf{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

$$\textbf{Precision} = \frac{TruePositives}{TruePositives+FalsePositives} \quad (8)$$

$$\textbf{Recall} = \frac{TruePositives}{TruePositives+FalseNegatives} \quad (9)$$

$$\textbf{F1} = 2 * \frac{recall * precision}{recall + precision} \quad (10)$$

2) Learning Curve: A learning curve graph is a plot that shows the performance of a model over a period of time. Learning curves are widely used in machine learning for algorithms that learn incrementally over time, such as deep learning neural networks. Reviewing a learning curve during training can be used to diagnose problems with learning, such as underfitting, overfitting, and whether the training and validation datasets are representative. As we can observe in Figure 11, we created two dual learning curves based on both the error and accuracy metrics. The learning curve of the train set (i.e. blue line) gives an idea of how well the model is learning whereas, the validation learning curve (i.e. orange line) gives an idea of how well the model is generalizing.

3) Class Activation Maps (CAM): A class activation map of a particular subject shows the distinctive image regions used by a CNN to identify that specific class. Given a simple connectivity structure and using a global average pooling layer, the weights of the output layer can be extracted. The weights are then projected back onto the convolutional feature maps to identify the importance of each region (e.g. using a heat map). This technique called activation mapping is beneficial

in showing whether or not the model extracts the right key regions from an image feature space. [27].

C. Implementation Details

1) Features: Firstly, we have extracted around 40 MFCC features. Even though the recommended number of coefficients is around 13-20, we have observed that going above 20 really helped the model's generalization capabilities.

2) Multi-task Model: For training the model we have used 5 kfold validation with a split of 20%. The model trained over 300 epoch, with a batch-size of 32, on the training and validation sets.

3) Task Weights: In multi-task learning, where multiple loss functions are present, we can statically choose how much each function counts towards the final loss by assigning a specific weight. When only classifying Wheezing and Rhonchus, we have kept the same weights for both tasks as the importance of each sound is not known. Therefore, a weight of 1 was assigned to both tasks. To visualize this, we can look at equation (3) where it shows how the final weighted loss is calculated based on the wight of each classification task

$$\textbf{loss} = 1 * wheezing\_loss + 1 * rhonchus\_loss \quad (11)$$

On the other hand, when classifying the survival rate, Wheezing and Rhonchus tasks were converted into auxiliary tasks. In other words, a small weight of $1e-1$ was assigned to the auxiliary task, while the main survival task weighted 0.8 (12). This way, we made sure to keep the Wheezing and Rhonchus tasks small enough to help the model better differentiate between very uncertain cases while, at the same time, making sure it does not influence our model in e negative manner to a point of over fitting.

$$\textbf{loss} = 0.1 * wheezing + 0.1 * rhonchus + 0.8 * survival \quad (12)$$

D. Experimental setup

In order to investigate the reliability of a CNN multi-task architecture, two models were trained, one for each experiment. The first model was used for the classification of MFCC features and a non-trained model using a transfer learning architecture was used to classify PCEN Mel Spectrograms.

1) Experiment 1: Wheezing and Rhonchus Prediction: The first experiment aimed to verify whether or not Rhonchus and Wheezing can benefit from a shared environment. As multi-task learning comes with perks like inductive transfer and implicit data augmentation this was a grate opportunity to prove the intrinsic relationship between these two abnormal sounds. Furthermore, we treated both tasks equally such that they count in the same proportion to the final result as we can not clearly tell which task is more important.

2) Experiment 2: Survival Prediction: In contrast to the previous experiment, this aimed to test the viability of our model when it comes to survival prediction. In this experiment, we kept the survival binary task as our primary task and transformed the Rhonchus categorical and Wheezes binary classifications into auxiliary tasks to investigate the possible correlation between these two sounds and the survival rate of a seal. We followed the natural example on how a veterinarian would check whether a seals is on the verge of dying or not. This means that we first try and find whether Wheezing or Rhonchus exists and then we try and predict the survival of a seal based on the severity and existence of these two abnormal sounds.

V. Results and Discussions

This section contains an in-depth analysis of classification performance of each model described in the previous section. Each result was evaluated using the validation techniques discussed in Section 4.

A. Experiment 1: Wheezing and Rhonchus

In our first experiment, we have used two separate models on two different types of features. The average accuracy of a 5-kfold split together with the classification report can be seen in Table I. We can observe that, when trained on MFCC features, our custom model managed to obtain, on average, an accuracy of 75% for Wheezing classification and around 63% when predicting Rhonchus. Furthermore, Figure 9 and Figure 10 show the confusion matrices of these two classifications. In Figure 9 we can observe a good differentiation between audio files that contain Wheezing and audio files that do not, by only missing 20% samples from the first class and 30% from the second one. In Figure 10, it can be seen that the Rhonchus prediction managed to predict 87% out of the severe cases correctly. Furthermore, it seems the model tends to sometimes classify Mild cases of Rhonchus as Ok, with a 27% chance to do so.

Figure 11 shows the training vs validation accuracy. The bottom image in the same figure shows the weighted loss of the model. It can be observed that the graph shows a steady increase in accuracy on both Wheezes and Rhonchus, with a minimal level of overfitting. On the other hand, the loss graph presents a continuous decrease in the weighted validation loss of both tasks. Even if both graphs show a jagged learning curve, we can clearly observe that the model gets better over time at predicting Wheezing and Rhonchus sounds.

In comparison to our first approach and also observed in Table II, using PCEN Mel-spectrograms decreased the accuracy of our model by 25% in case of Wheezing and 22% when predicting Rhonchus. One main reason for this is that the PCEN technique is too strong for our specific case. In Figure 7, it can be seen how most of the Wheezing present within the noisiness of the image is removed. This means that in the case of very subtle Wheezing and Rhonchus sounds, the PCEN method might remove them from the representation. The second big problem has to do with the zero-padding done in the pre-processing phase. Looking a the class activation map (CAM) in Figure 5, we can see that our model uses the black area at the end of the spectrogram as a region of interest (ROI) when classifying the audio files. The main reason the model uses this empty area is that the audio signals with 30 seconds do not have this in their representation. Therefore, the

model thinks that the zero-padded samples are unique in some form even though they are not. We tried solving this problem by trimming all the audio clips to the global minimum length that an audio file can have. However, this did not show any signs of improvement as the effects of the PCEN techniques had been already too damaging for our model to handle.

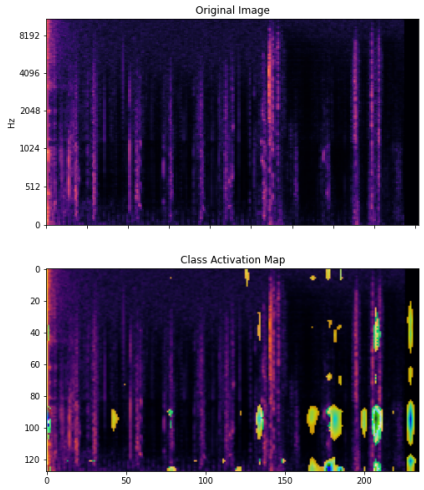

Fig. 7: The top image represents the original mel-spectrogram with per-channel energy normalization applied. The bottom image shows the class activation map (CAM) extracted from the last convolutional layer of our model. It can be observed how the zero-padded area at the end of the spectrogram is highlighted by the CAM.

We have also experimented with MFCCs as input to the pre-train ResNet50 model. However, this did not work, as these pre-trained models usually require input images with a size of at least (32, 32). Unfortunately, the mel-cepstral coefficients did not meet the required size to be used as input to the ReseNet50 model.

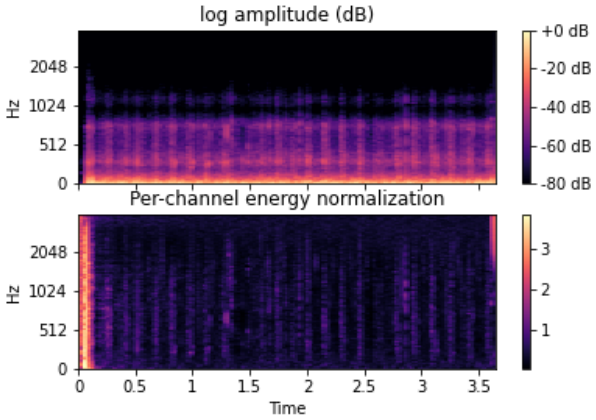

Fig. 8: First images represents the original mel-spectogram extracted from a very noisy audio signal in which subtle Wheezing and Rhonchus are present. The bottom image show how the PCEN technique destroyed all the noise including the subtle abnormal sounds present within that noise.

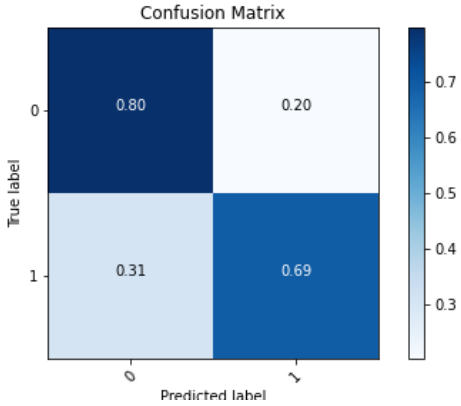

Fig. 9: The confusion matrix obtained from predicting the test split. A value of 0 means that the Wheezing sound is not present whereas a value of 1 shows the presence of this sound in the audio clip. The values in each cell were reported as percentages.

B. Experiment 2: Survival Prediction using Wheezing and Rhonchus as auxiliary tasks

Compared to the previous experiment, this one aimed to optimize the survival prediction of an animal. In this experiment we used whistling and Rhonchus as auxiliary tasks while survival remained our main priority. This means that a weight of 0.1 has been assigned to both Rhonchus and Wheezing losses whereas survival was given a weight of 0.8.

In Table II we can see the results of our experiment. Our binary survival classification obtained an accuracy of 80% while the accuracy of Wheezing and Rhonchus, which is lower compared to our previous experiment, stagnated at around 60%. However, this behaviour is natural, as the only role of Wheezing and Rhonchus tasks in this experiment was to introduce the inductive bias, implicit augmentation and attention focusing needed by the survival classification to improve its generalization capabilities and accuracy. Additionally, looking at the precision, recall and f1 score, we can see that the model has a low rate of incorrectly classifying samples, which further proves the generalization capabilities that our architecture obtained during the training phase.

The confusion matrix in Figure 12 shows that only one audio file, labeled as alive, was miss-classified by our custom CNN model. Although, the imbalance present in our survival class distribution is pretty pronounced, with only 22% out of the total samples being classified as dead, the model still managed to obtain 70% accuracy when classifying audio files related to dead seals. This further shows the efficiency of implicit data augmentation given by the addition of Wheezes and Rhonchus auxiliary tasks.

TABLE I: Wheezing And Rhocnhus Classification Results

Class	Model	Feature	Accuracy	Precision	Recall	F1
Wheezing	Custom CNN	MFCC	**0.75 ± 0.015**	0.75	0.73	0.76
Wheezing	ResNet50	PCEN	0.5	0.4	0.5	0.5
Rhonchus	Custom CNN	MFCC	**0.63 ± 0.025**	0.65	0.65	0.65
Rhonchus	ResNet50	MFCC	0.43	0.4	0.4	0.4

Table containing the average accuracy of each feature together with their classification reports. The MFCC features were trained on a custom model and PCEN were trained on a pre-trained ResNet50 model using transfer learning. The deviation of all folds was also reported in the accuracy column

TABLE II: Survival Classification Results

1 Class	Accuracy	Precision	Recall	F1
Survival	**0.80 ± 0.04**	0.93	0.83	0.67
Wheezing	0.68 ± 0.03	0.6	0.6	0.6
Rhonchus	0.52 ± 0.03	0.5	0.46	0.46

Table containing the average accuracy of our surival classification together with its classification report. For this experiment only the MFCC features and the custom CNN model were used.

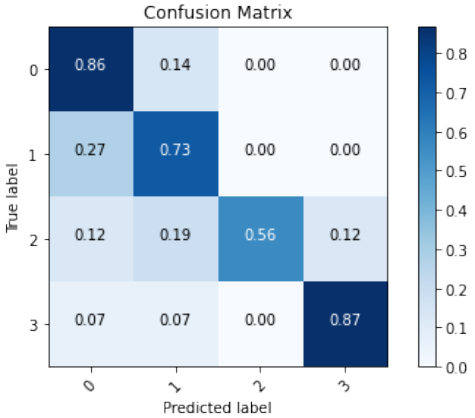

Fig. 10: The confusion matrix of Rhonchus classification, obtained from predicting the test split. Each class represents a severity level, starting at 0 which means Ok, 1 means mild, 2 is moderate and 3 represents a severe case

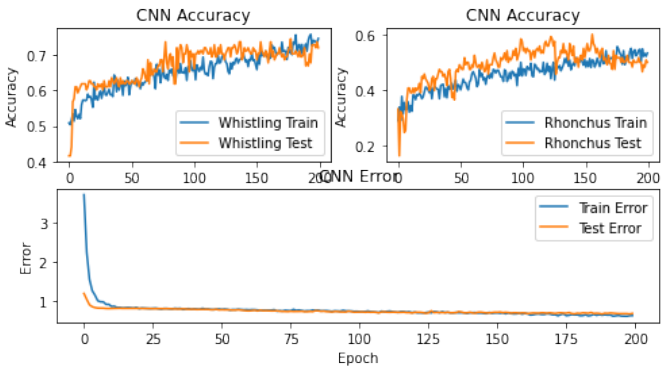

Fig. 11: The learning curve of our Wheezing And Rhonchus classification. The top two images show how individually each task performed in a shared environment. The bottom images shows the error loss fro med from the addition of the two loss functions of Wheezing and Rhonchus tasks

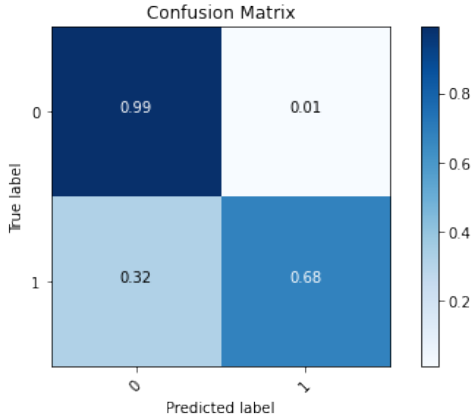

Fig. 12: The confusion matrix of survival classification, obtained from predicting the test split. A value of 0 shows that the animal will survive whereas a value of 1 means death

C. Discussion

1) Multi-task Architecture: In this paper we have touched on two types of MTL designs, with the main one being hard parameter sharing. This architecture has shown to work well when the classification tasks are closely related, which is the case with the Wheezing and Rhonchus sounds present in our data-set. Sonorous Wheezes (or Rhonchus) are Wheezing sounds that contain a snoring-like pattern, therefore, similar (and sometimes even identical) patterns emerge in both the spectral representations of both Wheezing and Rhonchus. This piece of information lead us to our hypothesis about the potential existence of a strong connection between these two abnormal sounds. By following this idea, we have built a model, that managed to capture the shared information between tasks and use it to correctly diagnose the types of pulmonary sounds present in the provided auscultations. The obtained results further prove the potential of MTL approaches when applied to problems such as mammals lung sound classification.

On the other hand, even though our survival classification did good, it would benefit more from a hierarchical approach such as soft parameter sharing. Our method uses a simple architecture in which different weights are assigned based on the type of task we are trying to optimize. However, using soft parameter sharing, we can actually mimic, to a certain extent, the diagnoses process conducted by veterinarians, which has the following steps:

- Detect any abnormal sounds.
- Give a presumptive diagnosis.
- Do more experiments.

Using the MTL approach, we are interested only in the first two steps, the detection and the presumptive diagnosis. Detection is more ore less given by the Wheezing and Rhonchus classification while the presumptive diagnosis can be seen as the survival prediction. Using MTL and soft parameter sharing, we could build a hierarchical model that first detects abnormal sounds and then, using the gathered information, tries to come up with a presumptive diagnosis for the survival chances of a seal.

2) Multi-task learning for lung classification: The overall results of this paper show that the classification of lung sounds through means of multi-task learning is not only possible but a very robust and powerful techniques which can easily detect the common features between different types of abnormal sounds. We demonstrated that, using state-of-the art models such as ResNet50, or custom built networks, one can obtain significant results with very limited data using the mel-cepstral coefficients. We have also shown the downsides of using methods such as per-channel energy normalization (PCEN) spectrograms for the classification of abnormal lung sounds.

While all of the research papers done in the field of lung sound classification using deep learning only use different kinds of single-task approaches, we can compare both the MTL and the single-task classification based on our obtained results. In [] Murat Aykanat et al. showed how machine learning algorithms such as CNN and SVM can accurately classify and pre-diagnose respiratory sounds by obtaining an accuracy of 80% using a CNN architecture on spectrogram-like images. In their study, they created a data-set of around 17,930 audio clips consisting of both healthy and sick subjects. In comparison to their work, our data-set only consisted of 146 labeled samples. Despite this huge difference in sample sizes, our model still managed to obtain very comparable results, with our best architecture having a difference in accuracy of only 5%, further proving the advantages of MTL when classification task are closely related.

VI. Conclusion and Feature Work

This paper evaluated the performance of multi-task learning and deep networks on the classification of audio signals extracted from seals lungs. Two classifiers were trained and tested on two different audio features, with the end goal being to see whether or not sharing information between related lung audio signals can improve our model's performance and generalization capabilities .

The results have shown that the custom model, using the MFCC features achieved the highest accuracy with a 73% chance to correctly classify Wheezing and 63% in the case of Rhonchus. Furthermore, using Wheezing and Rhonchus as auxiliary tasks proved our main hypotheses with regards to the strong relation between these two abnormal sounds and the death of a seal. Using MTL, we managed to introduce the inductive bias, attention focusing and implicit data augmentation that the model needed to correctly extract the intrinsic information found in the shared environment, in order to accruately classify the death of an animal based on its lung condition. However, the big imbalance found in the survival data-set makes it hard to tell whether this method is reliable enough to be applied on real subjects or real life scenarios.

On the other hand, the ResNet50 multi-task architecture using PCEN as input failed to give satisfactory results. Compared to our first approach, this technique decreased the accuracy of Wheezing classification by 25% and 22% in the case of Rhonchus.

This paper conisted of an overall investigation on the classification of lung sounds through means of MTL. While our custom built model did good, we still have used the hard parameter sharing approach. This technique was proposed by Caruana [28] 20 years ago and it is still the norm today. However, the main issue with this type of architecture is that it quickly breaks down if tasks do not share a lot of common characteristics or in case they require reasoning on different levels [11]. Recent works have shown that, rather than limit our model to squeeze all the task related knowledge into the same parameter spaces, is better to learn a task hierarchically. Drawing on the advances in MTL we could actually show our model how the tasks should interact with each other. Some popular examples of architectures that we would like tot try in order to improve upon our current work are, but not limited to: Deep Relationship Networks [29], Fully-Adaptive Feature Sharing [30], Cross-stitch Networks [31], Weighting losses with uncertainty [32] and Sluice Networks [33].

Acknowledgment

The author would like to thank Dr. Estefanía Talavera Martínez for her guidance ,help and feedback.

References

[1] N. Sengupta, M. Sahidullah, and G. Saha, "Lung sound classification using cepstral-based statistical features," *Computers in biology and medicine*, vol. 75, pp. 118–129, 2016.

[2] R. J. Callan, "Thoracic auscultation and percussion 2002 savma symposium."

[3] K. Lehnert, J. Raga, and U. Siebert, "Parasites in harbour seals (phoca vitulina) from the german wadden sea between two phocine distemper virus epidemics," *Helgoland Marine Research*, vol. 61, no. 4, pp. 239–245, 2007.

[4] S. Swarup and A. N. Makaryus, "Digital stethoscope: technology update," *Medical devices (Auckland, NZ)*, vol. 11, p. 29, 2018.

[5] D. Emmanouilidou, E. D. McCollum, D. E. Park, and M. Elhilali, "Computerized lung sound screening for pediatric auscultation in noisy field environments," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 7, pp. 1564–1574, 2017.

[6] T. Wang, Y. Shi, W. Hardt, Q. Feng, and L. Kang, "Heart sound acquisition with ecm sensor array and array data fusion algorithm." *International Journal of Performability Engineering*, 2020.

[7] A. Kandaswamy, C. S. Kumar, R. P. Ramanathan, S. Jayaraman, and N. Malmurugan, "Neural classification of lung sounds using wavelet coefficients," *Computers in biology and medicine*, vol. 34, no. 6, pp. 523–537, 2004.

[8] L. Chen, X. Mao, Y. Xue, and L. L. Cheng, "Speech emotion recognition: Features and classification models," *Digital signal processing*, vol. 22, no. 6, pp. 1154–1160, 2012.

[9] J. H. Hansen and G. Liu, "Unsupervised accent classification for deep data fusion of accent and language information," *Speech Communication*, vol. 78, pp. 19–33, 2016.

[10] Y. S. Abu-Mostafa, "Learning from hints in neural networks," *Journal of complexity*, vol. 6, no. 2, pp. 192–198, 1990.

[11] S. Ruder, "An overview of multi-task learning in deep neural networks," *CoRR*, vol. abs/1706.05098, 2017. [Online]. Available: http://arxiv.org/abs/1706.05098

[12] P. Gong, J. Ye, and C. Zhang, "Robust multi-task feature learning," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 895–903.

[13] X. Zhu, H.-I. Suk, L. Wang, S.-W. Lee, D. Shen, A. D. N. Initiative *et al.*, "A novel relational regularization feature selection method for joint regression and classification in ad diagnosis," *Medical image analysis*, vol. 38, pp. 205–214, 2017.

[14] Y. Zhu, X. Zhu, M. Kim, D. Shen, and G. Wu, "Early diagnosis of alzheimer's disease by joint feature selection and classification on temporally structured support vector machine," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2016, pp. 264–272.

[15] M. M. Van Hulle, "Self-organizing maps." 2012.

[16] J.-C. Chien, H.-D. Wu, F.-C. Chong, and C.-I. Li, "Wheeze detection using cepstral analysis in gaussian mixture models," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2007, pp. 3168–3171.

[17] T. Nguyen and F. Pernkopf, "Lung sound classification using snapshot ensemble of convolutional neural networks," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, 2020, pp. 760–763.

[18] S. İçer and Şerife Gengeç, "Classification and analysis of non-stationary characteristics of crackle and rhonchus lung adventitious sounds," *Digital Signal Processing*, vol. 28, pp. 18–27, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1051200414000384

[19] D. Bardou, K. Zhang, and S. M. Ahmad, "Lung sounds classification using convolutional neural networks," *Artificial intelligence in medicine*, vol. 88, pp. 58–69, 2018.

[20] Y. Zeng, H. Mao, D. Peng, and Z. Yi, "Spectrogram based multi-task audio classification," *Multimedia Tools and Applications*, vol. 78, no. 3, pp. 3705–3722, 2019.

[21] R. Kondor and S. Trivedi, "On the generalization of equivariance and convolution in neural networks to the action of compact groups," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2747–2755.

[22] S. Mallat, "Understanding deep convolutional networks," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150203, 2016.

[23] V. Lostanlen, J. Salamon, M. Cartwright, B. McFee, A. Farnsworth, S. Kelling, and J. P. Bello, "Per-channel energy normalization: Why and how," *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 39–43, 2018.

[24] L. Duong, T. Cohn, S. Bird, and P. Cook, "Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 845–850. [Online]. Available: https://www.aclweb.org/anthology/P15-2139

[25] J. Baxter, "A bayesian/information theoretic model of learning to learn via multiple task sampling," *Machine learning*, vol. 28, no. 1, pp. 7–39, 1997.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[27] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[28] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[29] T. Evgeniou, C. A. Micchelli, M. Pontil, and J. Shawe-Taylor, "Learning multiple tasks with kernel methods." *Journal of machine learning research*, vol. 6, no. 4, 2005.

[30] Y. Lu, A. Kumar, S. Zhai, Y. Cheng, T. Javidi, and R. Feris, "Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5334–5343.

[31] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3994–4003.

[32] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7482–7491.

[33] S. Ruder12, J. Bingel, I. Augenstein, and A. Søgaard, "Learning what to share between loosely related tasks."

Cristian Mihai Rosiu

Estefania Talavera Martinez