

EE INDUSTRIAL PLACEMENT: INTERIM REPORT

EZGİ ÖZYILKAN

ezgi.ozyilkana17@imperial.ac.uk CID: 01358595

Supervisor: DENİZ GÜNDÜZ

Company: IPC LAB, IMPERIAL COLLEGE LONDON

Line Manager: MIKOŁAJ JANKOWSKI

Submission Date: JULY 2, 2020

Imperial College London

CONTENTS

1	Introduction	3
1.1	About the Company	3
1.2	About the Lab	3
2	Project	3
2.1	Introduction	3
2.2	Methods	4
2.3	Results and Discussion	7
2.4	Future Progress	9
2.5	Required Technical Skills and Level of Responsibility	10
3	Project Planning and Management	10
3.1	Gantt Chart	10
3.2	Contingency Considerations	12
3.3	Reflections on Non-Technical Skills	12
4	Appendix	13

1 INTRODUCTION

In this section, the company and the lab where I have been undertaking my industrial placement is introduced. Research themes of interest for the lab are also highlighted.

1.1 About the Company

Imperial College London is a public university based in UK. It is established in 1907 by royal charter, combining the Royal College of Science, the Royal School of Mines and the City & Guilds College into one institution [1]. Being a polytechnic university, Imperial solely focuses on science, technology, medicine and business. With more than 59% undergraduate and postgraduate students coming from outside of the UK [2], Imperial enjoys a fairly diverse community. Imperial's student body is 39% female [3] and Imperial has recently won the Athena SWAN award, which recognizes practices advancing women's career in science, technology, engineering and mathematics [4].

1.2 About the Lab

Information Processing and Communications (IPC) Lab is a research team under Intelligent Systems and Networks (ISN) Group in Electrical Electronics Engineering Department at Imperial College London. The main focus of ISN Group is on how to build *autonomous* and robust systems (e.g. computers, networks, robots) which are adept at operating in complex, fast-paced environments [5]. IPC Lab is concerned with information theoretic privacy and security, compression, storage, efficiency aspects of information processing, as well as transmission over noisy channels and networks [6]. The lab has comprehensive experience in areas of multi-user information theory, wireless and energy harvesting communication networks, along with information theoretic analysis of noisy databases [6].

2 PROJECT

In this section, the project is described along with the technical details and the evaluation of experimental setup. The recent results obtained are also provided in Section 2.3.2. The technical skills applied are discussed in Section 2.5.

2.1 Introduction

The number of Internet of Things (IoT) devices has been growing expeditiously across the globe. From wireless cows [7] to smartwatches, the applications of IoT devices are quite diverse as well as unprecedented. As of 2019, the number of businesses that make use of IoT technologies has increased from 13% in 2014 to about 25% [8]. Equipped with reliable connectivity, high-tech sensors, more computing power and greater data-storage capacity, the growth of IoT technologies is expected to accelerate even further.

Currently, most IoT devices operate as wireless sensors at the **edge**¹: They collect data in order to pre-process and then offload it to an edge (or cloud) server. One of the main challenges in this setting is that the IoT devices are typically power and memory constrained — meaning they can only carry out a limited portion of the computations, which makes the resource allocation problem interesting. In our case, the computations of interest are imposed by a deep neural network (DNN). In this work, we will be specifically interested in multi-view cameras, whose fields of vision are overlapping with one another, to carry out person classification task at the wireless network edge². Additionally, since the cameras and the edge server communicate through a (wireless) channel, this transmission establishes a significant bottleneck as well as leading to latency and consumption of considerable energy by the resource-constrained cameras.

Some recent works [10], [11], [12] demonstrate the potential of splitting DNNs between the edge devices and the server. The practical objective of splitting the network as such is to make sure that the first part employed on IoT devices accommodates for their inherent limited computational and storage resources.

¹ Edge computing is a distributed computing paradigm where the computation workload is moved towards where the data is collected. It thus aims to reduce latency, bandwidth and overhead for the centralized data center [9].

² Given an image, we would like to identify the people appearing in that image. This is going to be achieved by a DNN that outputs a multi-hot encoded vector.

Furthermore, although Shannon's *separation theorem* allows separate design of source and channel coding, the joint source and channel coding (JSCC) is known to improve the performance and robustness in practical communication systems [13]. Considering real-time constraints, this makes deep JSCC scheme attractive for realistic distributed learning and inference scenarios such as the one we are concerned with.

Starting this project, our initial goals were the followings:

- Finding the optimal DNN splitting point between multi-view cameras and the edge server.
- Compressing the features to be sent to the server *even further* since the contents of the images are correlated, due to the nature of the cameras of interest³.
- Finding a large scale, good quality and annotated multi-camera dataset in order to deploy multi-view detection methods.

2.2 Methods

In this section, we propose a deep JSCC transmission scheme for multi-view cameras at the wireless network edge for the task of person classification. Considering the nature of surveillance applications, stringent bandwidth values will be considered — this means that features, instead of the original images, will be transmitted over the wireless channel. Since extracting such feature maps increases the computational load on edge devices, we'll be aiming to balance the trade-off between on-device computations and communication overhead. The section concludes with a 3-step training strategy for the entirety of the proposed system.

2.2.1 Proposed Architecture

Inspired by [12] and [14], we propose a **joint** classification scheme at the wireless edge (see Figure 1), which includes an autoencoder-based network for intermediate feature maps compression combined with deep JSCC scheme. By applying edge detection schemes onto multi-nodes of the network, we aim to reduce channel bandwidth requirements, compared to a single node setup. For the classification baseline, we employ the pretrained ResNet18 for each camera followed by three fully-connected layers, having Batch Normalization (BN) and rectified linear unit (ReLU) activation layers added after each of them (see Figure 2).

For the autoencoder architecture, we propose the one shown in Figure 3. This model choice aims to compress feature maps in order to reduce communication requirements as well as the on-device computational load, thanks to its asymmetrical structure (similar to the one discussed in [12]). The convolutional layers are either followed by Batch Normalization (BN), Generalized Divisive Normalization (GDN) or Inverse Generalized Divisive Normalization (IGDN) [15], [16] for the sake of Gaussianizing the data. We particularly employ GDN/IGDN layers since they are shown to be suitable for density modelling [15], [16] and are also used in the state-of-the-art image compression schemes (such as [17]). An additive white Gaussian noise (AWGN) channel is incorporated between the encoder and the decoder. Formally, let $\mathbf{X} \in \mathbb{R}^B$ and $\mathbf{Y} \in \mathbb{R}^B$ be the channel input and channel output vectors, respectively. The channel output is then calculated as $\mathbf{Y} = \mathbf{X} + \mathbf{Z}$, where $\mathbf{Z} \in \mathbb{R}^B$ is the noise vector drawn from $\mathbf{Z} \sim \mathcal{N}(0, \sigma_{\text{noise}}^2)$. Accordingly, the channel SNR is defined as:

$$\text{SNR} = 10 \log_{10} \left(\frac{P_{\text{signal}}}{\sigma_{\text{noise}}^2} \right) \text{ (dB)} \quad (1)$$

We also investigate sending the feature maps *separately*. Referring to this as **separate** device-edge model, an overview of the architecture is provided in Figure 4.

2.2.2 Training Strategy

The training strategy for the proposed joint model in Figure 1 consists of three steps. Firstly, we pretrain the classification baseline in Figure 2 with cross-entropy loss for 30 epochs, using SGD with Nesterov momentum of 0.9, learning rate of 0.01 and L_2 penalty weighted by $5 \cdot 10^{-4}$. The learning rate is reduced by a factor of 0.1 every 10 epoch.

³ For a more formal construction, let us say that $\mathbf{x}_1, \mathbf{x}_2$ is a pair of images and let $H(\mathbf{x}_1, \mathbf{x}_2)$ be their joint entropy. Given that the contents of these images are correlated due to overlapping fields of vision of the multi-view cameras, we have:

$$I(\mathbf{x}_1, \mathbf{x}_2) = H(\mathbf{x}_1) + H(\mathbf{x}_2) - H(\mathbf{x}_1, \mathbf{x}_2) > 0$$

Motivated by this observation, we seek a compression model to jointly compress the features of these correlated images.

⁴ Note that the use of the word *joint* in this case does not refer to the JSCC scheme. From now on, we will use this word to refer to both concepts and we will not be explicit as long as the context dictates its meaning.

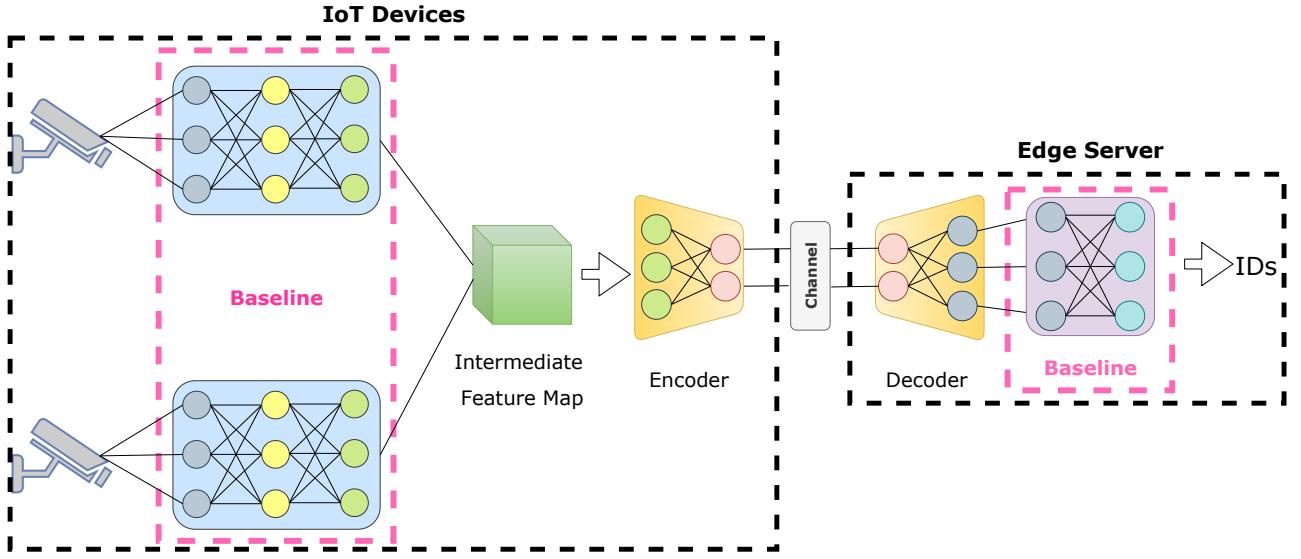


Figure 1: An overview of the proposed **joint** device-edge model. The baseline is split between the IoT devices and the edge server. The intermediate feature map generated at the splitting point is compressed and sent through the wireless channel by an autoencoder (see Figure 3), similar to the one discussed in [12]. At the server side, the received feature map is decoded and the edge server completes the forward pass in order to predict the unique person IDs. We will refer to this model as the *joint* model in the following of this report⁴.

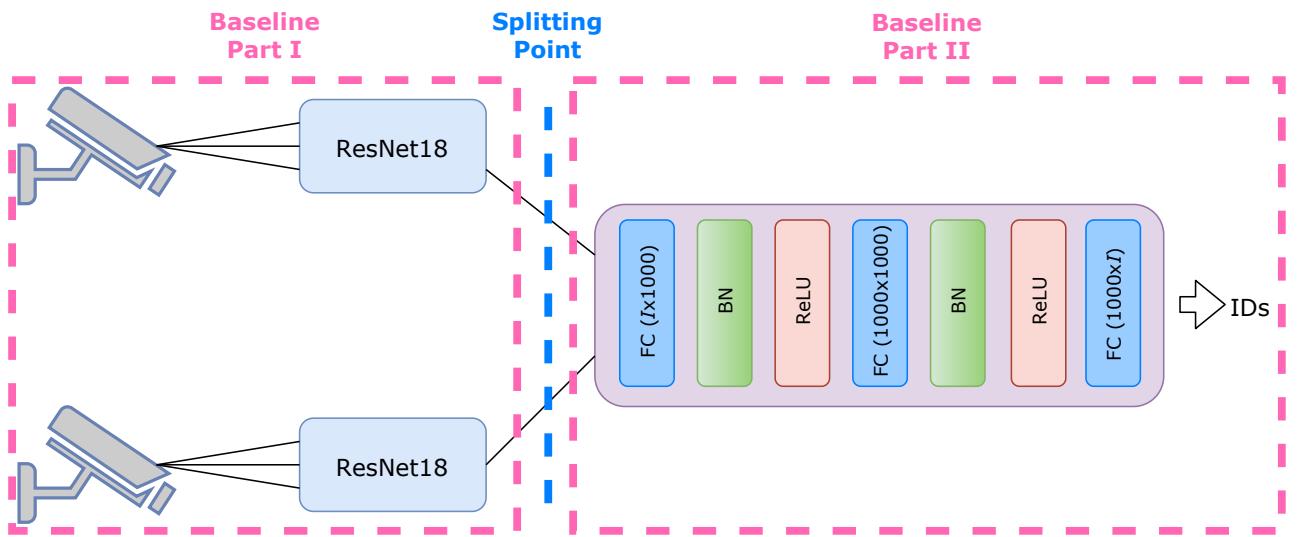


Figure 2: Proposed baseline architecture for classification. Fully-connected layer parameters are denoted as: input size \times output size. For two-camera setting, I in Figure 2 corresponds to $I = I_1 + I_2$, where I_K is equal to the number of the unique person IDs at Camera K .

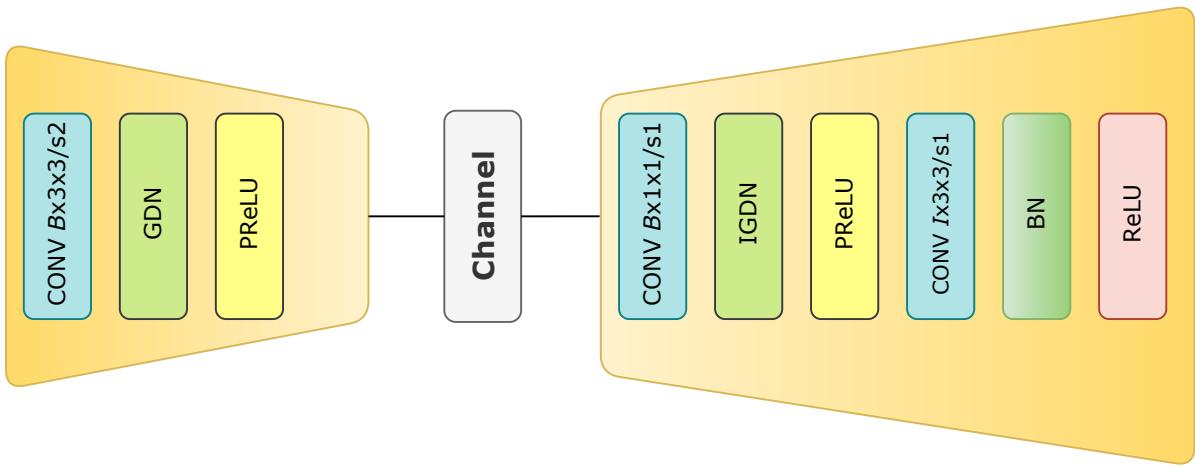


Figure 3: Proposed autoencoder architecture for the JSCC scheme. Its asymmetrical structure aims to reduce the computational load on the IoT devices (e.g. cameras in our case), similar to the design choice discussed in [12]. Convolutional layer parameters are denoted as: number of output channels \times kernel height \times kernel width / sampling stride (e.g. /s2 means sampling with stride 2). B , I and PReLU in Figure 3 correspond to channel bandwidth, input dimension and parametric rectified linear unit (PReLU), respectively. Note that I in Figure 2 and Figure 3 refer to the same variable.

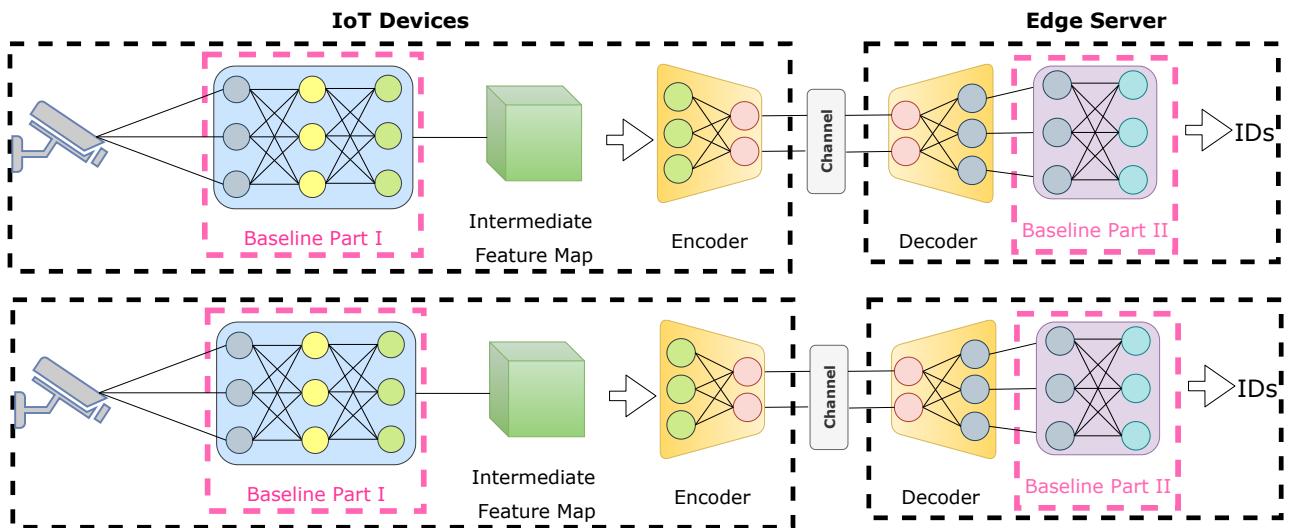


Figure 4: An overview of the **separate** device-edge model. Similar to the joint model in Figure 1, the baseline is split between the IoT devices and the edge server at the same splitting point indicated in Figure 2. The intermediate feature map is compressed and sent through the wireless channel by the same autoencoder architecture in Figure 3. Note that in Figure 4, the autoencoders are independent of each other and the channels incorporated are orthogonal with respect to one other. The total channel bandwidth for the separate model is calculated as $B = B_1 + B_2$, where B_K is the bandwidth allocated for Camera K . We will refer to this as *separate* model in the following of this report.

Secondly, each set of images in the training set is forward passed through Baseline Part I, shown Figure 2, in order to extract all the possible intermediate feature maps at the splitting point. These feature maps are then used to pretrain the autoencoder in Figure 3 with L_1 loss for 50 epochs, using SGD with Nesterov momentum of 0.9, learning rate of 0.1 and L_2 penalty weighted by $5 \cdot 10^{-4}$. The learning rate is reduced by a factor of 0.1 after 20th and 40th epochs. Note that AWGN channel model is incorporated during training of the autoencoder so that the autoencoder is able to learn the robust transmission of feature maps.

Thirdly, the entire network in Figure 1 is trained end-to-end by combining Baseline Parts I and II, and placing the pretrained autoencoder at the splitting point shown in Figure 2. Similar to the first step, the network is trained with cross-entropy loss for 50 epochs, using SGD with Nesterov momentum of 0.9, learning rate of 0.01 and L_2 penalty weighted by $5 \cdot 10^{-4}$. Likewise, the learning rate is reduced by a factor of 0.1 every 10 epoch.

Instead of single-step training of the joint model *directly*, the proposed multi-step training strategy allows the joint model to achieve superior classification accuracy, even with significantly low channel bandwidths.

The training strategy for the separate model in Figure 4 is same as the 3-step training discussed for the joint model, along with the identical L_2 penalty and learning rate reduction applied for the baseline, autoencoder and entire network.

2.3 Results and Discussion

In this section, we evaluate the performance of the joint and separate models shown in Figure 1 and 4, respectively. Before presenting the results, the experimental setup along with the motives for the dataset choice is discussed first.

2.3.1 Experimental Setup

To assess the performance of the proposed joint model, we use the labeled ‘WILDTRACK’ dataset [18], where there are 7 static cameras whose fields of view are overlapping within one another⁵. One of the main characteristics of the ‘WILDTRACK’ dataset is that these 7 static cameras capture the challenging and realistic setup of walking pedestrians in front of the main building of ETH Zurich, Switzerland [18]. Furthermore, the high precision joint-camera calibration and synchronization of the ‘WILDTRACK’ dataset surpasses those of the PETS 2009 [19], which has been recognized as a challenging benchmark dataset at the time of publishing. Although the EPFL-RLC [20], another multi-view dataset, improves the joint-calibration accuracy and synchronization of multi-camera setup compared to the PETS 2009, it provides annotations only for a small subset of the total frames, making it unsuitable for deep-learning-based multi-view detection schemes [18].

The ‘WILDTRACK’ dataset provides annotations for 400 synchronized frames for each 7 static camera, using a frame rate of 2 fps. Note that original resolution of the images is 1920×1080 and on average, there are 20 pedestrians on each frame [18]. Due to unbalanced nature of labels in the ‘WILDTRACK’ dataset, the loss function for classification task in the baseline and the entire model is **cross-entropy with logits**⁶, which aims to optimize the metric

$$\text{Balanced Accuracy} = \frac{\text{Sensitivity} + \text{Specificity}}{2} = \frac{\text{TPR} + \text{TNR}}{2} \quad (2)$$

for all class predictions, where TPR and TNR stand for True Positive Rate and True Negative Rate, respectively.

In order to ensure that one needs to effectively transmit the intermediate feature maps coming from both cameras in Figure 1, we define a correlation metric, r_{corr} , to choose which pair of overlapping cameras to use for the evaluation of the proposed scheme⁷:

$$r_{\text{corr}} = \frac{1}{N} \sum_{i=1}^N \frac{|\text{Cam}_{k,i} \cup \text{Cam}_{j,i}|}{|\text{Cam}_{k,i}| + |\text{Cam}_{j,i}|} \quad (3)$$

In Equation (3), $\text{Cam}_{k,i}$ refers to the set of people appearing at Camera k for ith frame and note that $N = 400$ for the ‘WILDTRACK’ dataset. After calculating this correlation metric for all combinations of 2 cameras from the ‘WILDTRACK’, we found that the pair of Camera 4 and Camera 5 has the greatest r_{corr}

⁵ Annotated dataset can be downloaded from: <https://www.epfl.ch/labs/cvlab/data/data-wildtrack/>

⁶ We used the PyTorch implementation of BCEWithLogitsLoss, which is documented at: <https://pytorch.org/docs/stable/nn.html>

⁷ Clearly, we pick two distinct cameras to calculate this metric: $j \neq k$ in Equation (3).

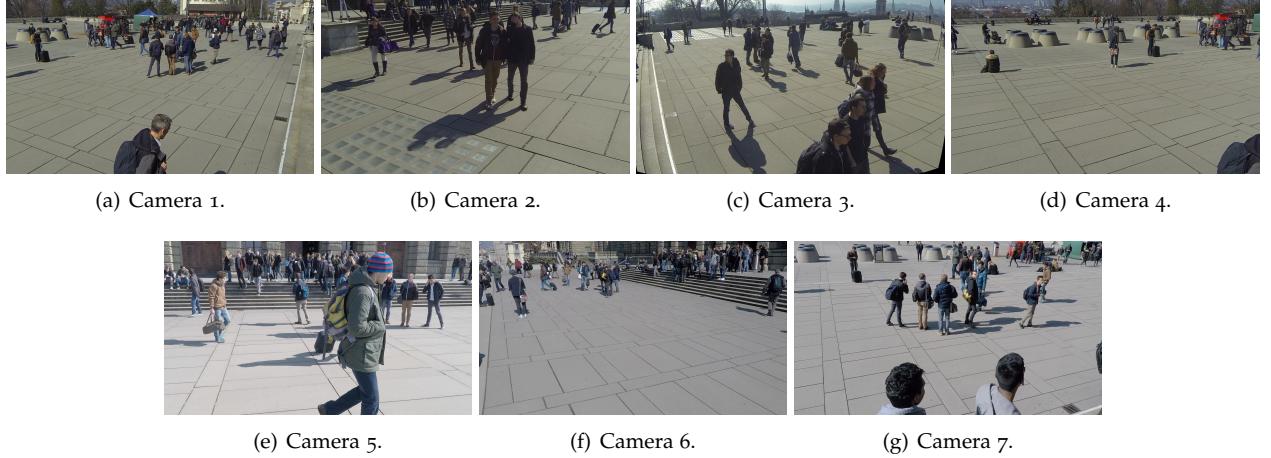


Figure 5: Synchronized corresponding frames from 7 static cameras of the ‘WILDTRACK’ dataset [18]. Four GoPro Hero 3 and three GoPro Hero 4 cameras are used for frames shown in the first and second row, respectively [18]. As seen from Figure 5, the data recording took place under sunny weather conditions and the height of camera positions is above average human height. The fields of vision of the cameras are also noticed to be overlapping as discussed in Section 2.1 (e.g. the concrete stairs in Camera 5 and Camera 6). Occlusions between pedestrians are also observed (e.g. Camera 5) just as the ‘WILDTRACK’ captures a densely crowded realistic scenario in front of the university ETH Zurich.

value of 0.911⁸. Therefore, we choose Camera 4 and Camera 5 to measure the performance of the proposed joint model for the task of person detection via multi-label classification.

2.3.2 Progress to Date

In order to make a fair comparison between the joint and separate models, we need an accuracy metric to assess the performance of the both models at the evaluated bandwidths. For this end, we define a metric for the separate model, $\text{acc}_{\text{separate}}$, for a given bandwidth budget of b as the following⁹:

$$\text{acc}_{\text{separate}}(b) = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \max_{b_1, b_2 \in \mathcal{B} \text{ s.t. } b_1 + b_2 = b} (w_{\text{Cam}_4} \cdot \text{acc}_{\text{Cam}_4, b_1} + w_{\text{Cam}_5} \cdot \text{acc}_{\text{Cam}_5, b_2}) \quad (4)$$

$$\text{where } w_{\text{Cam}_4} = \frac{|\text{ID}_{\text{Cam}_4}|}{|\text{ID}_{\text{Cam}_4}| + |\text{ID}_{\text{Cam}_5}|}, \quad w_{\text{Cam}_5} = \frac{|\text{ID}_{\text{Cam}_5}|}{|\text{ID}_{\text{Cam}_4}| + |\text{ID}_{\text{Cam}_5}|},$$

\mathcal{E} : set of independent experiments,

\mathcal{B} : set of evaluated bandwidths,

$\text{acc}_{\text{Cam}_k, b_z}$: mean balanced accuracy across all classes for Camera k at bandwidth b_z ,

ID_{Cam_k} : set of all people appearing at Camera k

This metric ensures that the configuration of separate model has the flexibility to choose the best bandwidth allocation given a bandwidth budget of b . We will refer to this metric as *Best Combination* in the following of this report.

We plot the mean balanced accuracy of the joint and separate models as a function of channel bandwidth in Figure 6, Figure 7, Figure 10 and Figure 11¹⁰, and as a function of channel SNR in Figure 8. Results in Figure 6 and Figure 7 clearly show that the joint model beats the *Best Combination* for separate model at almost all bandwidths for SNRs 20 dB and 0 dB. We can also observe that the similar accuracy is achieved at

⁸ Since

$$r_{\text{corr}} = \frac{1}{N} \sum_{i=1}^N \frac{|\text{Cam}_{k,i} \cup \text{Cam}_{j,i}|}{|\text{Cam}_{k,i}| + |\text{Cam}_{j,i}|} = \frac{1}{N} \sum_{i=1}^N \frac{|\text{Cam}_{k,i}| + |\text{Cam}_{j,i}| - |\text{Cam}_{k,i} \cap \text{Cam}_{j,i}|}{|\text{Cam}_{k,i}| + |\text{Cam}_{j,i}|}$$

having greater r_{corr} value indicates having on average less number of people simultaneously appearing at Camera k and Camera j .

⁹ Getting the weighted average is equivalent to computing the accuracy of the entire separate model. This is useful because there is no need to train a new model for each combination of bandwidths; it suffices to train the individual networks for both cameras independently. This means we only need to train $2|\mathcal{B}|$ times instead of $|\mathcal{B}|^2$.

¹⁰ For Figure 10 and Figure 11, please refer to the Appendix.

a smaller bandwidth with the joint model for almost all accuracy values. This supports the main hypothesis that the features sent to the edge server can be *further* compressed by jointly compressing and decompressing the feature maps coming from multi-view cameras whose fields of vision are overlapping. Furthermore, it can be observed from Figure 8 that the joint model outperforms the separate one for indicated SNRs.

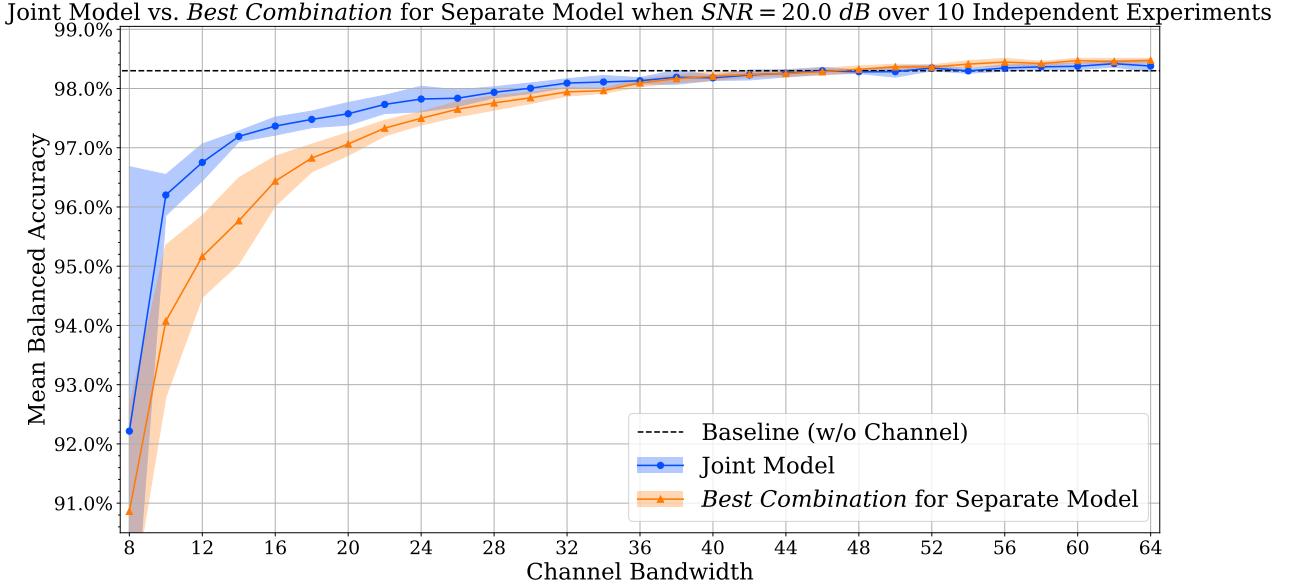


Figure 6: Mean balanced accuracy as a function of channel bandwidth for $\text{SNR}_{\text{train}} = \text{SNR}_{\text{test}} = 20 \text{ dB}$, where $B \in \{4, 6, 8, 10, \dots, 64\}$ for the separate model in Equation 4.

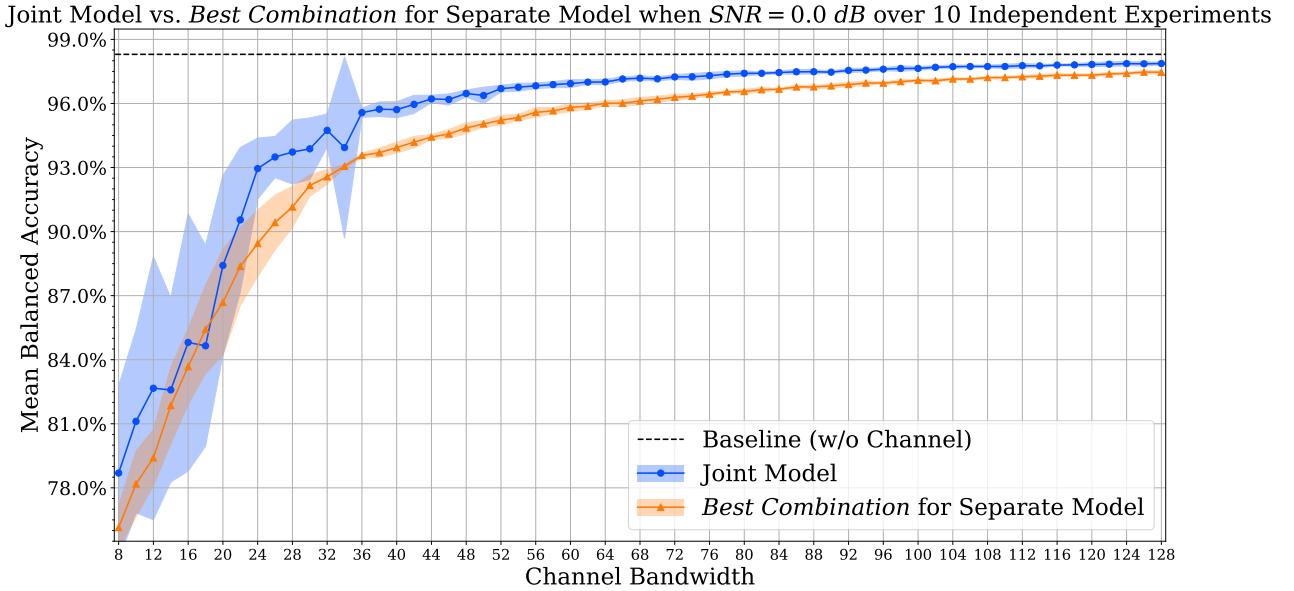


Figure 7: Mean balanced accuracy as a function of channel bandwidth for $\text{SNR}_{\text{train}} = \text{SNR}_{\text{test}} = 0 \text{ dB}$, where $B \in \{4, 6, 8, 10, \dots, 128\}$ for the separate model in Equation 4.

2.4 Future Progress

We aim to provide a comprehensive evaluation of the proposed joint model architecture in Figure 1 by:

- Using various channel SNRs and numerous bandwidth constraints

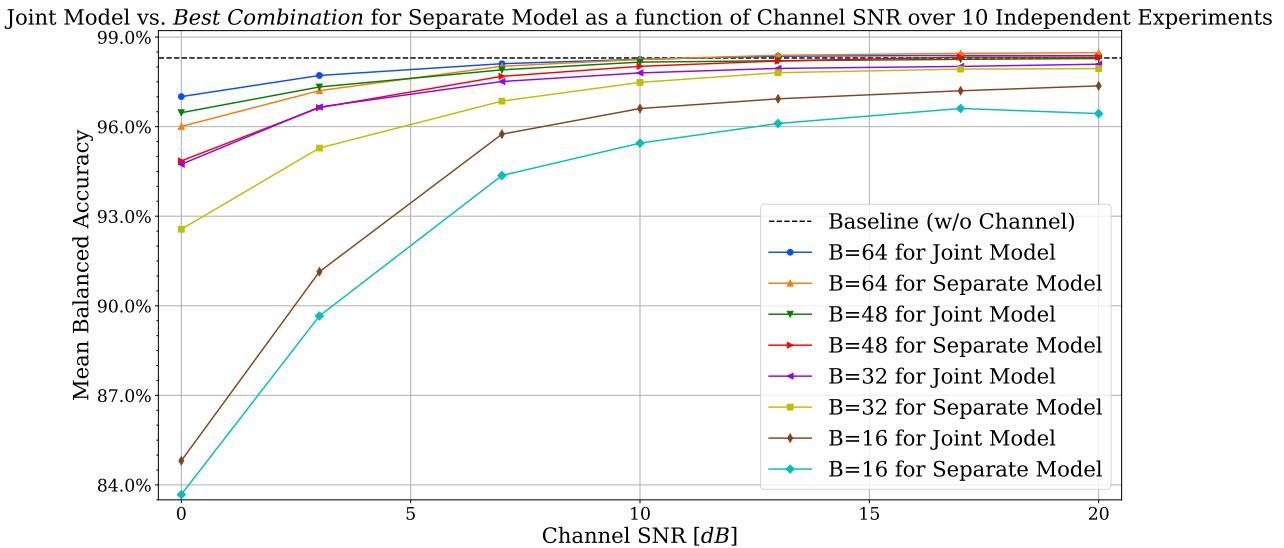


Figure 8: Mean balanced accuracy as a function of channel SNR, where $P_{\text{signal}} = 1$ and $\sigma_{\text{noise}}^2 \in \{1, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01\}$ in Equation 1.

- Comparing with different channel types (e.g. non-orthogonal¹¹ and/or fading channels) and model configurations other than the separate one in Figure 4 (e.g. having 2 separate encoders and 1 shared decoder)
- Choosing different sets of overlapping cameras from the 'WILDTRACK' dataset
- Comparing with respect to digital (separate) transmission schemes

Also find the project planning for the remainder of the placement in Figure 9.

2.5 Required Technical Skills and Level of Responsibility

To implement a deep JSCC scheme in order to carry out person classification at the wireless network edge, one needs to have sufficient knowledge in the following fields:

- Machine Learning and Deep Learning
- Communication and Information Theory

Since IPC Lab has published some recent works [12],[14] using JSCC approach over wireless channels, my starting point was based on the proposed architecture in [12]. Because the source code of the scheme was implemented in PyTorch [21], which is a machine learning library based on Python that is used to implement neural networks with strong GPU acceleration, I had to familiarize myself with Python and PyTorch, especially with the implementation of *modular* neural networks¹².

In order to remotely access to IPC Lab's servers so that I could train the neural network models, I got familiar with some essential command-line tools such as ssh, tmux and vim.

As I wanted to make my project open source, I also published a GitHub repository¹³, where I regularly push the revised source code and write about ongoing experiments and recent results.

3 PROJECT PLANNING AND MANAGEMENT

3.1 Gantt Chart

See the Figure 9.

¹¹ For an overview of this scheme, please refer to Figure 12 in the Appendix.

¹² Opposed to a single large neural network, each module in a *modular* neural network is assigned to a specific task and ultimately connected with each other in order to produce a final solution.

¹³ <https://github.com/ezgimez/Multi-label-classification>

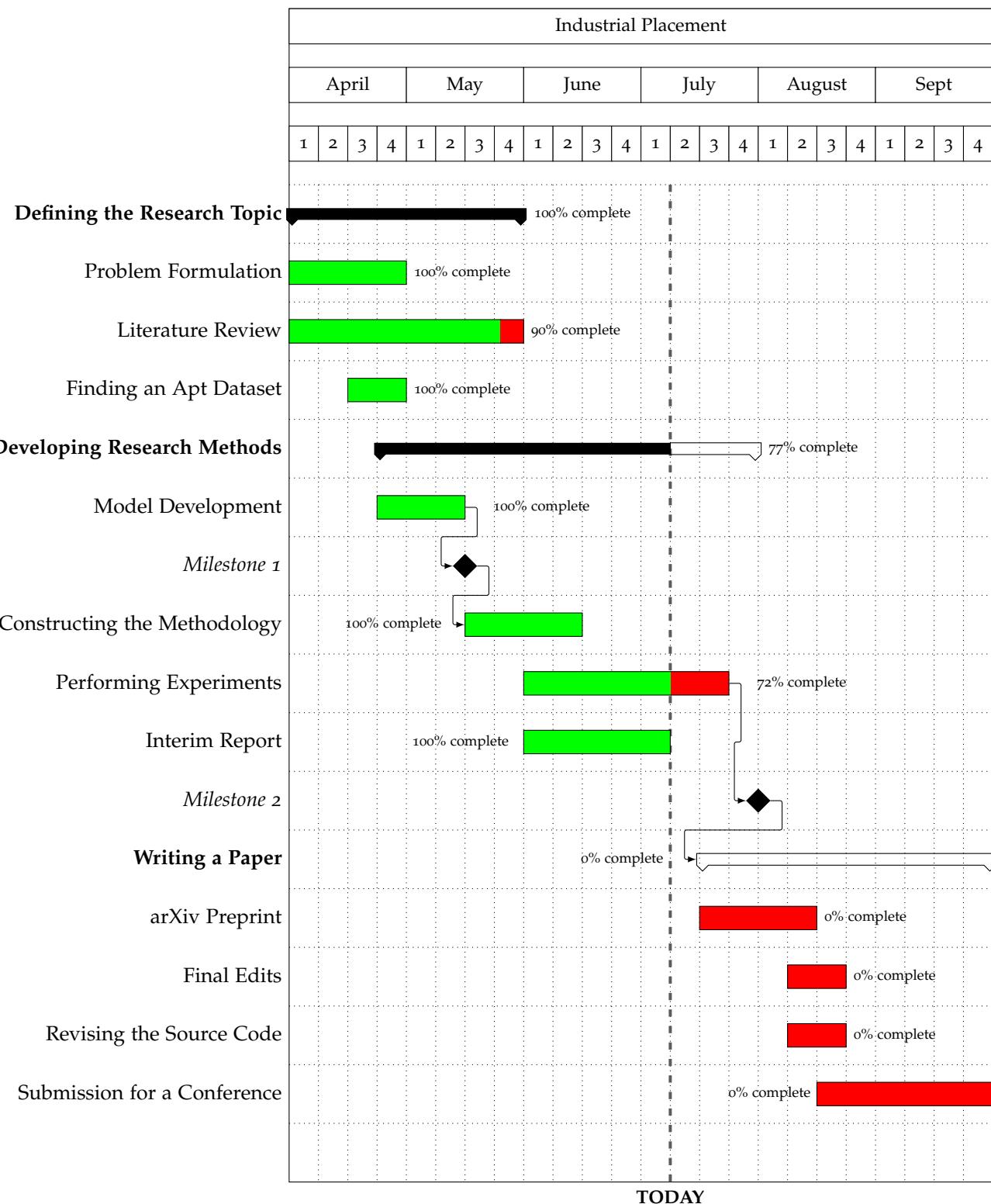


Figure 9: Gantt Chart for the Industrial Placement.

3.2 Contingency Considerations

If the empirical evidence didn't support the main hypothesis discussed in Section 2.1, that is overlapping images from multi-view cameras can be *further* compressed before being sent to the edge server, I would be taking the following approaches:

- Changing the proposed joint device-edge model architecture shown in Figure 1
 - Examining various DNN splitting points, other than the one in Figure 2
 - Trying different layers for Baseline Part II shown in Figure 2
 - Employing different autoencoder models for the JSCC scheme
- Adopting a different learning strategy, other than the one discussed in Section 2.2.2
 - Altering the hyperparameters employed in the multi-step training strategy
 - Foregoing pretraining phases for the baseline and the autoencoder
- Trying different set of cameras as input from the 'WILDTRACK' dataset

3.3 Reflections on Non-Technical Skills

Working in a research lab has its own perks: (possibly) having flexible working hours, being immersed in a mentally challenging environment, surrounded by collaborative and knowledgeable people, constantly learning new things as part of the job (e.g. through conferences, group meetings), being able to work on high-risk ideas that are well ahead of the state-of-the-art in the industry... However, it can end up being quite lonely and may even lead to an unbalanced lifestyle as it is not a monotonous nine-to-five job where one regularly interacts with other colleagues day-to-day. On the other hand, as one (generally) has immense degree of freedom over the daily schedule, it is both feasible to wake up at 12:00 every morning and work 70 hours per week.

One of the non-technical skills that I've improved during my time at IPC Lab is time management. Although I had struggled to set up a routine in the beginning, I believe I managed to create a daily routine consisting of:

- Planning a list of experiments to perform, upcoming meetings, papers to read
- Communicating latest results to my line manager and discussing next steps with him

According to my own experience, it was particularly useful to keep a **consistent communication** between my line manager, Mikołaj, and myself. Mikołaj has effectively advised me on which type of experiments to conduct, how to record and present the results and how to write the relevant explanations in a clear, unambiguous manner, which helped me to improve my academic writing skills.

Even though there is no employer-imposed working hours in place, I found it useful to regularly follow a self-imposed schedule. Unlike my previous job experience in a financial firm, the degree of autonomy expected from me during my research placement has turned out to be a liberating, and equally challenging, experience as I was the one in charge of the direction of my own work.

As a result of the Covid-19 pandemic, I also have had a chance to develop work-from-home skills. Since millions of workers may never go back into the office as regularly as they were used to [22], I believe work-from-home skills will be one of the in-demand skills for the future. I think the post-pandemic era will also witness a change in the dynamics of success criteria in the workplace — just as conferences, in-person meetings, handshakes might not worth the risk of infection, those who have results-oriented mindset are more likely to prevail whereas those who thrive on face-to-face interactions are doomed to struggle [23]. During my work-from-home period, I managed to create my own dedicated workspace in order to balance my blurrier-than-ever work and home routines. Setting boundaries with my family (e.g. not being disturbed between 11:00-20:00) and also with my work (e.g. not binge-working till 1:00 in the morning) helped me to appreciate the flexibility of working from home while keeping me focused during the day.

4 APPENDIX

Joint Model vs. Best Combination for Separate Model when $SNR = \infty$ dB over 10 Independent Experiments

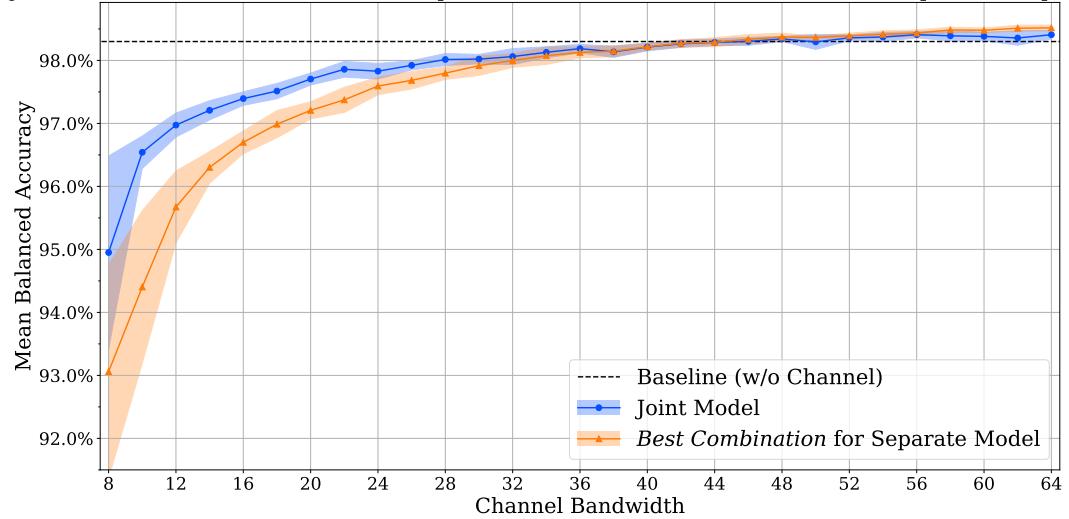


Figure 10: Mean balanced accuracy as a function of channel bandwidth for $SNR_{train} = SNR_{test} = \infty$ dB, where $B \in \{4, 6, 8, 10, \dots, 64\}$ for the separate model in Equation 4.

Joint Model vs. Best Combination for Separate Model when $SNR = 10.0$ dB over 10 Independent Experiments

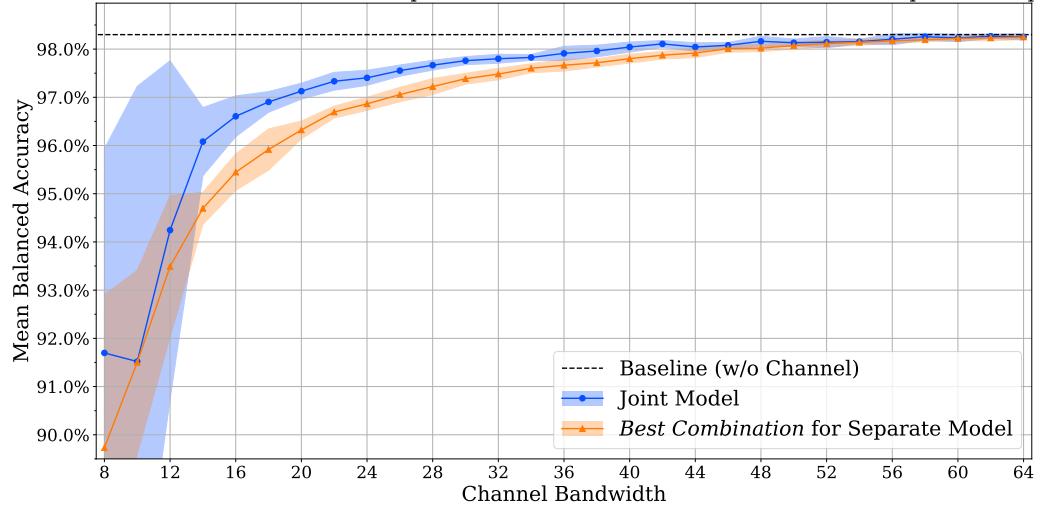


Figure 11: Mean balanced accuracy as a function of channel bandwidth for $SNR_{train} = SNR_{test} = 10$ dB, where $B \in \{4, 6, 8, 10, \dots, 64\}$ for the separate model in Equation 4.

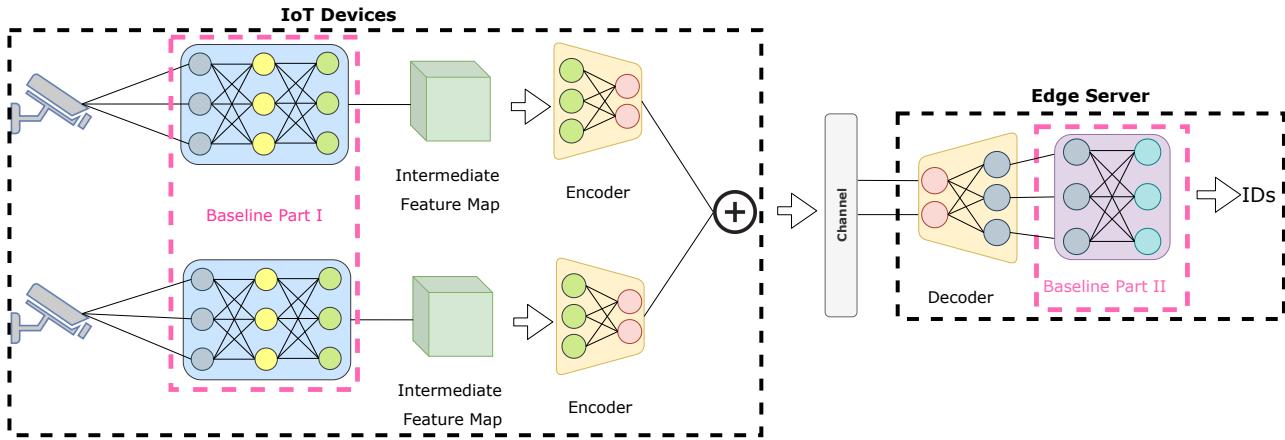


Figure 12: An overview of the non-orthogonal channel model. Similar to the joint model in Figure 1, the baseline is split between the IoT devices and the edge server at the same splitting point indicated in Figure 2. The intermediate feature map is compressed and sent through the wireless channel by the same autoencoder architecture in Figure 3. Unlike the joint model, 2 distinct encoders are employed which use the channel in a non-orthogonal fashion. The channel symbols for this model is implemented as $S = S_1 + S_2$, where $S_{\mathcal{K}}$ is the channel symbols for Camera \mathcal{K} .

REFERENCES

- [1] Imperial College London. *A timeline of College developments*, N.D. (Accessed May 28, 2020). <https://www.imperial.ac.uk/about/history/college-developments/>.
- [2] Wikipedia. *Imperial College London*, 2020 (Accessed May 28, 2020). https://en.wikipedia.org/wiki/Imperial_College_London#cite_note-14.
- [3] Higher Education Statistics Authority. *Where do HE students study?*, N.D. (Accessed May 28, 2020). <https://www.hesa.ac.uk/data-and-analysis/students/where-study#provider>.
- [4] Imperial College London. *Facts and figures*, N.D. (Accessed May 28, 2020). <https://www.imperial.ac.uk/about/introducing-imperial/facts-and-figures/>.
- [5] Imperial College London. *Intelligent systems and networks*, N.D. (Accessed May 28, 2020). <https://www.imperial.ac.uk/electrical-engineering/research/intelligent-systems-and-networks/>.
- [6] Imperial College London. *Research*, N.D. (Accessed May 28, 2020). <https://www.imperial.ac.uk/electrical-engineering/research/intelligent-systems-and-networks/research/>.
- [7] David Evans. *Introducing the wireless cow*, 2015 (Accessed June 4, 2020). <https://www.politico.com/agenda/story/2015/06/internet-of-things-growth-challenges-000098/>.
- [8] Alexander Rajko Fredrik Dahlqvist, Mark Patel and Jonathan Shulman. *Growing opportunities in the Internet of Things*, 2019 (Accessed June 4, 2020). <https://www.mckinsey.com/industries/private-equity-and-principal-investors/our-insights/growing-opportunities-in-the-internet-of-things#:~:text=The%20number%20of%20businesses%20that,almost%20threefold%20increase%20from%202018.&text=1>.
- [9] Eric Hamilton. *What is Edge Computing: The Network Edge Explained*, 2018 (Accessed June 4, 2020). <https://www.cloudwards.net/what-is-edge-computing/>.
- [10] Amir Erfan Eshratifar, Amirhossein Esmaili, and Massoud Pedram. *BottleNet: A Deep Learning Architecture for Intelligent Mobile Cloud Computing Services*. *arXiv e-prints*, page arXiv:1902.01000, February 2019.
- [11] Jiawei Shao and Jun Zhang. *BottleNet++: An End-to-End Approach for Feature Compression in Device-Edge Co-Inference Systems*. *arXiv e-prints*, page arXiv:1910.14315, October 2019.
- [12] Mikolaj Jankowski, Deniz Gunduz, and Krystian Mikolajczyk. *Deep Joint Transmission-Recognition for Power-Constrained IoT Devices*. *arXiv e-prints*, page arXiv:2003.02027, March 2020.

- [13] Eirina Bourtsoulatze, David Burth Kurka, and Deniz Gunduz. Deep Joint Source-Channel Coding for Wireless Image Transmission. *arXiv e-prints*, page arXiv:1809.01733, September 2018.
- [14] Mikolaj Jankowski, Deniz Gunduz, and Krystian Mikolajczyk. Deep Joint Source-Channel Coding for Wireless Image Retrieval. *arXiv e-prints*, page arXiv:1910.12703, October 2019.
- [15] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. Density Modeling of Images using a Generalized Normalization Transformation. *arXiv e-prints*, page arXiv:1511.06281, November 2015.
- [16] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end Optimized Image Compression. *arXiv e-prints*, page arXiv:1611.01704, November 2016.
- [17] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv e-prints*, page arXiv:1802.01436, January 2018.
- [18] Tatjana Chavdarova, Pierre Baqué, Stéphane Bouquet, Andrii Maksai, Cijo Jose, Louis Lettry, Pascal Fua, Luc Van Gool, and François Fleuret. The WILDTRACK Multi-Camera Person Dataset. *arXiv e-prints*, page arXiv:1707.09299, July 2017.
- [19] J. Ferryman and A. Shahrokni. Pets2009: Dataset and challenge. In *2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 1–6, 2009.
- [20] Tatjana Chavdarova and François Fleuret. Deep Multi-camera People Detection. *arXiv e-prints*, page arXiv:1702.04593, February 2017.
- [21] 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 H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [22] The Economist. *What will be the new normal for offices?*, 2020 (Accessed May 29, 2020). <https://www.economist.com/britain/2020/05/09/what-will-be-the-new-normal-for-offices>.
- [23] Olga Khazan. *Work From Home Is Here to Stay*, 2020 (Accessed May 29, 2020). <https://www.theatlantic.com/health/archive/2020/05/work-from-home-pandemic/611098/>.