A Deep Learning Approach On Fusion Technique Comparison Applied To Affordance Classification

Christophe Friezas Gonçalves
STUDENT NUMBER: 2059012

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR IN COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee:

M. Kirtay
G. Spigler

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
May 2023

Preface

This thesis was written to fulfill the graduation requirements for the bachelor Cognitive Science and Artificial Intelligence at Tilburg University. The project helped to explore a robotic interest of mine in an academic surrounding. It was an enjoyable yet a bit stressful experience, yet the final result still being a proud achievement of mine. A thank you to my supervisor dr. Murat Kirtay for the continuous feedback throughout my thesis project.

A Deep Learning Approach On Fusion Technique Comparison Applied To Affordance Classification

Christophe Friezas Gonçalves

Affordance and action recognition bring with them many different cavities and complex individualized implementations and techniques in robotics and computer vision. This study tackles a comparison set between feature level and decision level fusion, comparing simple implementations, to take one step back from highly complex architectures. The comparison confronts multimodal fusion on a visual based iCub action recognition dataset, while implementing a multi-tasking solution towards affordance classification in a robotic agent observer setting. All in an effort to solidify the use of simple modern tools and the foundation created for the aforementioned cavities in these diverse yet similar branches of AI, robotics and computer vision. Both fusion techniques apply convolutional neural networks evaluated on accuracy and precision, resulting in accuracies over chance level, capable and explainable representations incorporating the benefits each modality yields. Late fusion coming out on top with better accuracy and stability, encouraging the tendency of modern science in stacking and fusing highly optimized models on decision levels, thus fusing the beneficial properties of different models.

1. Introduction

A button, a simple and functional object in our everyday life. Pressing this button seems so simple for a human. We infer what it does by the context around and on it. An elevator button with a number on it for example. A different way, humans infer its purpose is by the knowledge we acquired over time while using said button, a button on a game controller for example. Both of these subsume inference capabilities we humans innately develop, robots on the other hand do not. Over the past years, this problem has had many breakthroughs, object affordance techniques have been developed and refined giving robots the ability to grasp objects and tools on their own (Hassanin, Khan, and Tahtali 2021). So let us take a step back and take a deeper look at the basics that made these breakthroughs possible. This study consequently focuses on the foundation of object affordances, fusion techniques and action recognition, to reinforce this foundation and help take two steps forward in tackling a complex task with said foundation.

A common practice in the world of action recognition is using RGB-D data. It combines color images with depth data. This mimics a similar human approach, our vision, we see in color and both eyes combined enable the brain to extract the depths before us (Parker 2007). These multimodal datasets enable models to encompass more information. An increase in information leads to an improvement in inference power and yet, with varied information comes the issue of how to tackle multiple different inputs. The models applied towards these problems vary from deterministic to probabilistic methods and more importantly for this study the models apply feature level,

decision level or intermediate fusion, of which the latter will not be tackled in this paper (Min et al. 2016).

Feature level fusion, also referred to as early fusion, fuses the different modalities into one big input feature to feed it into one network. Decision level fusion, referred to as late fusion, bases its technique on the results of multiple networks, one for each modality. Late fusion being preferred given the nature of science, models and techniques developed over time and from one model iteration to the next, leading to single, highly optimized and accurate models that only need to be connected on the decision level with others. This study aims to tackle this notion of fusing the models at their decision level and its improved properties.

The comparison is applied to a dataset intended towards human robot interactions, an idea that increases in popularity given the progress of Boston Dynamics. As an example, the more their humanoid and autonomous robots gain in capabilities, the more we integrate them in workplaces, improving efficiency and workflow in addition to increasing human safety. Hence a robot needs to achieve inference capabilities to work in unison with another human, extending the effectiveness and efficiency of the human agent. This thesis thus focuses on action and tool recognition to further develop the notion of bonding human and robot work. A robot with the capabilities of inferring tool use and action results improves work efficiency with a human or robot co-agent along with predicting dangerous actions and avoidance thereof.

Research Question:

To condense, the paper aims to compare two fusion techniques (early and late fusion) applied on the iCub action recognition dataset (Kirtay et al. 2020). The results aim to progress the field of robotic, AI and computer vision to focus on the best technique and improve human robot interaction along the way. The early fusion model will set our baseline and the late fusion model is our contestant. Both techniques are evaluated on accuracy and precision. The general and main research question will be subdivided into two sub questions to dive deeper into both techniques and the underlying models to examine them in detail. The research questions thus are:

Main RQ: What impact do different types of multimodal fusion have on the precision and accuracy using the iCub action recognition dataset to predict action and tool duos ?

Sub RQ1: How well does feature-level fusion (early fusion) perform on the iCub action recognition dataset using convolutional neural networks inferring action and tool duos?

Sub RQ2: How well does decision-level fusion (late fusion) perform on the iCub action recognition dataset using convolutional neural networks inferring action and tool duos?

1.1 Findings

The comparison of both techniques provides a deeper insight into the stability fusion provides regarding convolutional neural network architectures. A trend emerged in both models, showing elevated difficulty in classifying the tool class compared to the action class. The late fusion model achieving higher accuracies than the early fusion model in the overall comparison. In addition, the category precisions show higher stability for the late fusion model. The results in kind favouring the decision level implementation.

A note being that the performance of both models, reaching accuracies over 0.8, being high over chance level (0.25), proved the viability of both fusion techniques.

2. Related Work

The paper tackles, as mentioned before, multiple topics at one time. This gives way to examine the different branches on their own and some combined work throughout.

2.1 Object Affordance

Until now we threw the word affordance around without going too deep into its precise meaning. Affordances of an object, as originally defined by James J. Gibson, tell us what said object offers or which interactions it enables the user in its environment (Gibson 1979). Important to note is that an object may have multiple affordances. The notion of using affordance in robotics, computer vision and AI enables different directions in itself. Affordance categorization, segmentation, social and interactive affordances, to name a few (Hassanin, Khan, and Tahtali 2021). Affordance categorization is ordinarily used before many other methods: segmentation, for example. Nguyen et al. (2016) exhibit great results on an affordance detection task, revealing high promise on the implementation of convolutional neural network based architectures. A survey on visual affordance, on the other hand, shows a dominant focus on mathematical approaches when it comes to affordance classification and detection instead of deep learning based approaches (Hassanin, Khan, and Tahtali 2021). In addition the survey shows a lack of multimodal implementation in both categorization and detection (Hassanin, Khan, and Tahtali 2021).

2.2 Action Recognition

The recognition of actions seems straightforward, yet actions in themselves are divisible into different parts. Robots enact actions through hard coded scripts or through the help of AI, learned by demonstration, reinforcement learning and more. Herath, Harandi, and Porikli (2017) gave a comparison between different architectures and their application on static-image and video based action recognition. Their survey shows the superiority of deep neural networks in comparison to hand crafted representations (Herath, Harandi, and Porikli 2017). A consequence being a hard steer towards Deep learning architectures given their performance boosting abilities. From this point on we intertwine our branches, considering affordance playing a big role in action recognition and vice-versa. The studies named so far, which implement affordance based approaches, use datasets with clearly annotated actions for their objects or have pretrained models, trained on similar datasets, opening a direction into multi-tasking (Hassanin, Khan, and Tahtali 2021; Nguyen et al. 2016; Lakani, Rodríguez-Sánchez, and Piater 2018). In other words data with action labels allow us to simultaneously detect affordance if a tool or object is involved.

2.3 Multimodal Data and Fusion Techniques

The more complex the model or system applied by a machine or robot gets, the higher the information load. Multimodal models and tasks emerge from this phenomenon, given more motors for more degrees of freedom, more sensors to get a deeper understanding of the surroundings. To illustrate how swift modalities increase, an example

will be used. Let us take a self-driving car equipped with proximity sensors, a LiDAR scanning array, a RGB camera and GPS trackers. All these sensors have to be considered for optimal decision making, swiftly accumulating four different modalities. The adoption of using a color image and depth data (RGB-D) dataset is widely shared among robotic and computer vision experiments, depth adding an important notion on actions (Song et al. 2022; Nguyen et al. 2016; Chu et al. 2019). Bayoudh et al. (2021) show different approaches to said task and modern trends of fusion techniques, ending in the consensus that choosing the right data and fusion technique is crucial and not a given. A study by Kniesmeijer and Kirtay (2022) applies fusion on the same dataset as used in this paper, showing good performance for convolutional neural networks while incorporating fusion on an action recognition task. On that note, let us proceed to fusion. Decision level fusion and feature level fusion perform on an even basis if the correct fine tuning and pre-processing steps have been applied (Ramachandram and Taylor 2017). Recent papers show a trend towards late and intermediate fusion, given the aformentioned pretrained and optimized models (Herath, Harandi, and Porikli 2017; Joze et al. 2020; Song et al. 2022; Yao, Lei, and Zhong 2019). Science bases its progress on improving what came before, building upon older models and knowledge. Fine-tuning and improving, creating a logical tendency towards implementing late fusion, starting from scratch takes more time and financing in addition to lacking the necessary quality testing.

2.4 Scientific Aim

The current paper takes a mixed approach of the stated topics. The investigation lies in adopting a complex task with the basic tools of these branches. The idea is to solidify the foundation on which these branches and tools stand. Action recognition and affordance categorization performed by convolutional neural networks (CNNs), to incorporate the multi-tasking claim as well as the tendency towards CNN based architectures and their performance boosting potential. The main comparison between feature level (sub RQ1) and decision level fusion (sub RQ2), investigates the trend mentioned before about late-fusion preference and superiority. All together performed on an RGB-D robotic aimed dataset to situate the experiment in the robotic and computer vision field additionally adding multi-modal data for the fusion implementation.

3. Methods

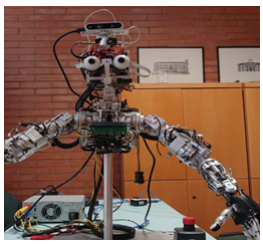
Let us now explore the data and architecture applied in our comparison. The start will be made with details about the iCub action recognition dataset followed by the pre-processing procedures, the structure and reasoning of the chosen architectures before ending with the evaluation metrics and software.

3.1 Dataset

The experiment uses the iCub action recognition dataset, created and collected by Kirtay et al. (2020) and publicly available online[1]. To give some background on the data, the iCub robot, depicted in Figure 1., used in the study, was upgraded with a depth sensor enabling the acquisition of depth data, left eye and right eye color images and a center

1 https://box.hu-berlin.de/d/6bc742f6dafb4dc2a36a/

Figure 1: Picture of the enhanced iCub robot



color camera. This study uses the data recorded from the center camera and the depth sensor.

The indicated dataset consists of images and depth data of 20 objects in their starting position and their end position, achieved after an action has been applied to the objects, with a specific tool. The iCub robot is set as an action observer while the action is executed by an agent opposite of it. The robot observing only two static states, no tool nor movement.

The set of actions consist of pulling, pushing, left to right movement and right to left movement, applied by a slingshot, spatula, ruler and a hook. The set of 20 different objects do not play a role in this study, their specifications thus being omitted. The object-action-tool triplets were repeated 10 times each, culminating in 25600 RGB images and depth data points in total, of which we will use 12800, given aforementioned dropping of both individual eye situated cameras.

3.2 Pre-processing and Splits

The images run through a pre-processing pipeline, resizing them to 64x64 pixels and transforming them into tensors (Programming specific data structure). For the normalization step, color images have to be differently normalized compared to depth data. The normalization occurs by calculating the z-score. In other words, the mean (γ) and standard deviation (σ) of each color channel for each image is calculated. Thereafter the z-score normalization formula (1.) was applied to each channel.

$$x_{norm} = \frac{x - \gamma_x}{\sigma_x} \qquad (1.)$$

The issue with depth data is the missing red channel for many of these data points, missing referring to the zero valued red channel. As a consequence the red channel has no need for normalization given the zero values along the entirety of the channel. Importantly, this did not apply to all the depth data points. In case of a standard deviation, higher than zero the channel was normalized as previously mentioned, applying equation (1.). At this point the way forward splits in two, for early fusion the RGB images and depth data were concatenated vertically into one big image of 64x256 pixels. The late fusion models need two input matrices, one for each modality, thus resulting in RGB images being concatenated vertically into one 64x128 matrix and a same sized depth data input matrix. Appendix A. illustrating the resulting input matrices before normalization, showing shape and order. Important to note is the order, the before

action image is situated on top of the after action image, repeated for each modality. For early fusion, the color images being situated on top of the depth images.

The data is split into 70% training (2240 entries), 15% validation (480 entries) and 15% test (480 entries) data for early and late fusion.

3.3 Architecture

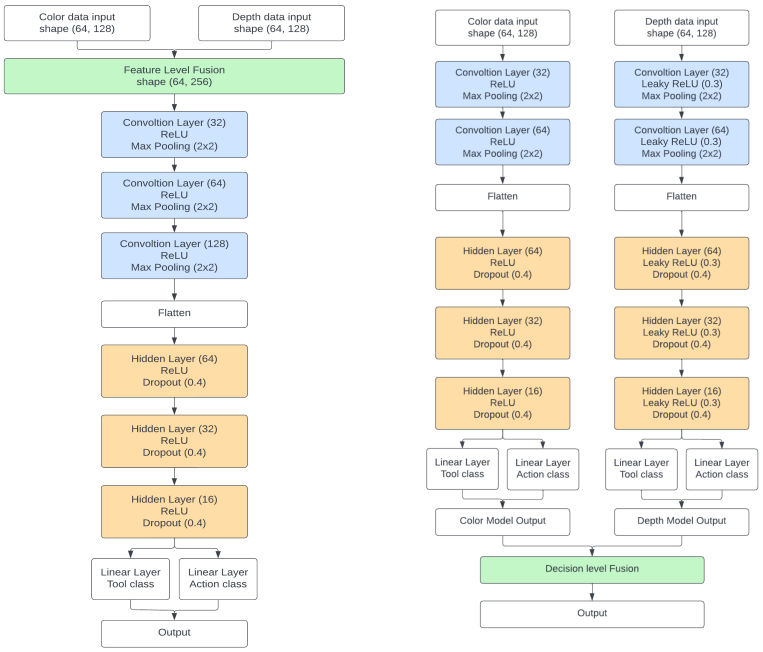
The study applies convolutional neural networks. Since their introduction in 1980 as a Neocognitron, CNNs evolved and showed high promise in the fields of robotics and computer vision, given their ability to extract spatial representations and underlying patterns from visual inputs (Bayoudh et al. 2021; Fukushima 1980). A CNN brings with it the benefit of three distinct properties, sparse connectivity, parameter sharing and equivariant representations (Goodfellow, Bengio, and Courville 2016). Sparse connectivity allows convolutions to diminish the size of saved parameters, reducing the memory load, increasing statistical efficiency and most importantly accelerating the output calculations given the aforementioned diminished parameter size. Parameter sharing emerges from the use of a kernel, in our case a 3 by 3 kernel. The entire input of a convolution layer is screened with said kernel, thus passing every location of the image with the same weights. There are exceptions given boundary values if we do not allow for padding (artificially expanding the image). In our case we pad the image by one outer layer. The last property, equivariant representations, is the most important one for our experiment. The property allows our model to extract features from our input in certain areas of our image, no matter if the features were translated (location shifted). The detected feature appears in the same shifted location. The equivariant notion tells us that convolution may come before the movement or vice versa, which gives our model the power to detect the features of an object before it was moved and after an action has been performed. Combining all three properties we end up with a model having an explainable representation of movements, actions and object features without sacrificing memory load and speed (Goodfellow, Bengio, and Courville 2016).

The basis for the models utilized were derived from the aforementioned study by Kniesmeijer and Kirtay (2022) consequently the models were changed, adapted and fine tuned to fit our aim. The early fusion model, applied in this study, consists of a CNN with three convolution layers having 32, 64 and 128 output filters respectively. Each convolution layer is followed by a Rectified Linear Unit (ReLU) activation layer, their key property lying in non-linearity springing from a strict non-zero policy. The output is flattened and proceeds to the three hidden layers, defined with 64, 32, 16 output filters followed each time with ReLU activation functions. The final layer, so to speak, consists of two individual fully connected layers, to separate the result classes, one layer for the tool class and one layer for the action class output. This split has been implemented to test the multi-tasking aspect of the models mentioned in the introduction.

The given outputs are evaluated with the cross entropy loss function. The selected loss function enables multi-class classification by applying a SoftMax activation and calculating the cross-entropy loss. Both losses, tool class and action class loss, are calculated individually and summed up before the backward pass. During the proceeding backward pass the indicated loss value updates the weights of our network, enabling the hidden layers to adapt and learn.

The late fusion models have similar layouts. The neural networks themselves are smaller, both having one convolution layer less, the last one being omitted (128 output filters). The color image network utilizes the ReLU activation function as seen in the early fusion model. The depth data network switches to leaky ReLU activation functions

Figure 2: Model illustrations. Feature level fusion (left) and decision level fusion (right) depicted with overfitting countermeasures and parameters for each layer. Batch Normalization omitted given application on all layers, except class layer and flatten layer



to counteract the dying ReLU effect, which emerged on training runs. As previously seen, one of the channels in the depth data is predominantly zero. This sets up a majority of our model to have no activation and for some exceptions in said channel, which have low values along the red channel, and thus have no impact on the model, given the none negative activation of ReLU and fixing the activation results on zero values. A leaky ReLU activation allows for a small negative slop, allowing said low values to impact the model. The slop parameter was set to 0.3. Figure 2. depicting a visualization of both fusion model architectures.

The models apply the Adam optimizer with a learning rate of 0.0001 to apply stochastic gradient descent on the backward pass. Adam was selected for both fusion techniques given its wide use and good performance (Ruder 2016) and the learning rate resulted from hyperparameter testing.

A cavity in AI is overfitting, the notion of a model training excessively and learning the training set by heart, correlating one specific input to one specific output, not learning any underlying patterns. Consequently ruining the generalization capabilities of the model. Overfitting was circumvented in our models by applying batch normalization, dropout in the hidden layers and early stopping in the training cycles. The batch normalization recentres the signals in between layers, to keep a normal distribution throughout the forward pass. It applies the same z-score normalization based on the mean and standard deviation as seen in the pre-processing section. Dropout is a method implemented in the hidden layers. As the name implies, the hidden layer drops certain neurons in the forward pass. This promotes the layers to develop overall representations

in each neuron, different connections from and towards the previous and proceeding layers and avoids neurons to co-adapt (Baldi and Sadowski 2013). In other words, it relies solely on one specific preceding neuron. The dropout rate was set to 0.4 for all models. The last countermeasure against overfitting is early stopping. A neural network is trained through epochs, an epoch being one training cycle, where the model pass over the whole train set once. A usual procedure is to set a fixed epoch amount and let it run until it reaches said epoch value. In our case we set the epoch value to a high number and apply a test criterion on the validation and training loss. Once the training loss reaches a lower value as compared to the validation loss we stop the training given that the model starts to overfit and our generalization ability is diminishing.

3.4 Evaluation

With the models set up, we validate their architectures with 10 fold cross-validation. Combining validation and training data performing 10 separate runs on the architecture while reshuffling training and validation data in addition to reinitializing the model on each run. From those runs we pick the best performing model and proceed to the testing. On this note, a clarification has to be made. Picking the best model has a very vague definition. The best model, in this case, being the best performing model based on the validation accuracy of the tool class results. The reasoning behind picking this metric is expanded upon in the discussion section. In the late fusion implementation the cross validation has been performed on the individual models, picking the best performing model for color and depth individually. After selecting said best models, one additional step had been made in the late fusion implementation, the decision level fusion itself. The final evaluation is made based on the average prediction of the color and depth model output. The evaluation metrics of this thesis consist of the overall accuracy (tool and action combined) in addition to a deeper dive into the separate tool and action class accuracies and precisions for the subsequent categories. The accuracy shows the overall rate at which our model predicts the correct output. The precision on the other hand tells us about the model specific predictive power for each category per class. The results are accompanied by confusion matrices, as a visual display of correct and incorrect predictions per category, allowing for emerging pattern observation.

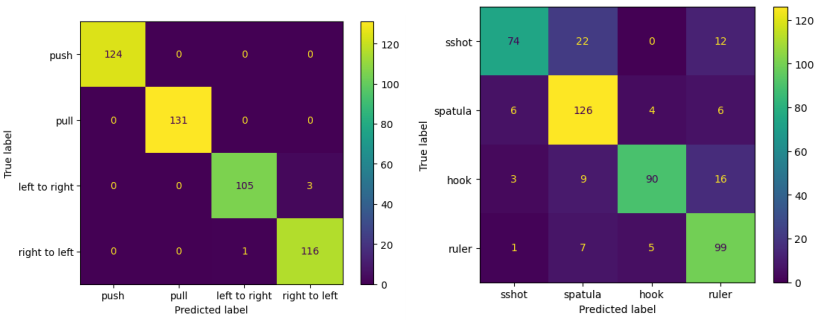
3.5 Software

The aforementioned Architecture in addition to the pre-processing has been performed in a Python environment (Version 3.9.16) (Van Rossum and Drake 2009). The Pillow (9.4.0), Torchvision (0.15.0) and NumPy (1.23.5) libraries enabled the pre-processing, loading, transforming and basic mathematical manipulations (Clark 2015; Falbel 2023; Van der Walt, Colbert, and Varoquaux 2011). Torch (2.0.0) brought the deep learning framework with dataset architectures expanded upon with Pandas (1.5.3) data frames (Paszke et al. 2019; McKinney 2010). Finally Scikit-learn (1.2.2) enabled cross-validation and splitting of the data and Matplotlib (3.7.1) the visualization of the input features and confusion matrices (Pedregosa et al. 2011; Hunter 2007)

4. Results

The next section explores the results acquired. The baseline is formed by the early fusion implementation and explored first, followed by the late fusion model and the

Figure 3: Confusion matrices of the action class (left) and tool class (right) for the early fusion model



comparison of both implementations. The chance level for both implementations sits at 0.25 for each class, tool and action, given the 4 categories per class.

Table 1: Accuracies and precisions achieved by the early fusion model

	Accuracy	Action	Push	Pull	Left to Right	Right to Left
Action	0.99	Precision	1.00	1.00	0.99	0.97
Tool	0.81					
Overall	0.9	Tool	Slingshot	Spatula	Hook	Ruler
		Precision	0.88	0.77	0.91	0.74

4.1 Early Fusion

The early fusion model as shown in Table 1. reached an overall accuracy score of 0.9, resulting in a performance over chance level. The individual classes show diverging predictive power on the accuracy metric. The action accuracy reached a near perfect prediction score, where on the other hand the tools stayed at 0.81, which is a strong predictive potential an interesting difference. The action class shows balanced precision throughout the categories, “Right to Left” being the least performant with 0.97. The tools, on the other hand, show a diverse picture. The “Hook” label achieving the best accuracy with 0.91 and “Ruler” the lowest with 0.74, resulting in a difference of 0.17 between the highest and lowest category. The tool class thus decreasing the overall accuracy.

The matrices in Figure 3. show, as previously stated, a near perfect prediction for each category, missing 3 in “Left to Right” and one in the opposite direction. Interestingly only confusing horizontal directions. The tools give a scrambled matrix. The most mistakes were made on predicting the slingshot tool, “Spatula” and “Ruler” labels being the common predicted label for 34 of the mislabel slingshots. This could occur based on the basic affordance and actions the dataset allows. A simple push from a ruler looks similar to a simple push from a spatula or slingshot if pushed with the tip.

4.2 Late fusion

The late fusion results presented are split into two tables, Table 2. presenting the accuracy scores for each model. Table 3. showing the category precision per model. The two individual models tackling the modalities and the fused model outputting the final prediction. The separation is performed in order to closely examine the contribution each model and modality brings to the fused results.

Table 2: Accuracy results achieved by the two individual and proceeding fused model

Accuracy	Color model	Depth model	Fused model
Action	0.99	0.97	0.99
Tool	0.73	0.78	0.85
Overall	0.86	0.875	0.92

The color model achieved a general accuracy of 0.86. Reaching a score of 0.99 on the action and 0.73 on tool accuracy. The precision for the action class reflecting a similar trend as the early fusion model, whereas the tool class reflecting a similar mixed bag. Lowest precision resulting in 0.65 and highest reaching 0.84. Accumulating a difference of 0.19 in between, a higher difference in comparison to the early fusion model. Consequently emerging from the missing information, the depth data added to the representation in the previous model. Figure 4. shows the confusion matrices for both individual models. A noticeable find on the color model matrix is the tendency of the model to classify a datapoint as a slingshot, having mislabelled 64 of the 480 as slingshot. Interestingly spatula and ruler being the correct label in the majority of the mislabelled cases, steering towards the same reasoning as proposed for the early fusion model.

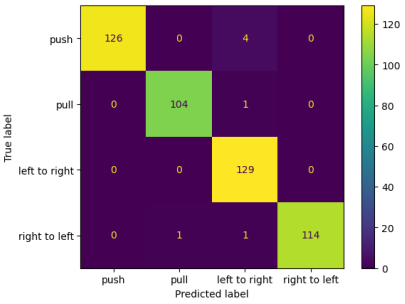
The depth data model in comparison performed worse on the action set, ending in an accuracy of 0.97. In the greater scheme of things, the difference appears small, yet noteworthy when taking the difference between the tool accuracies of both models into account. The depth data improves the tool prediction in contrast to the color model by

Table 3: Precision results achieved by the two individual and proceeding fused seperated per class

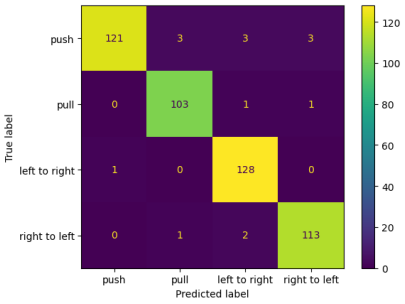
Action	Color model	Depth model	Fused model
Push	1.00	0.99	1.00
Pull	0.99	0.96	0.98
Left to Right	0.96	0.96	0.96
Right to Left	1.00	0.97	1.00
Tool			
Slingshot	0.65	0.83	0.82
Spatula	0.84	0.80	0.91
Hook	0.76	0.74	0.84
Ruler	0.72	0.73	0.81

Figure 4: Confusion Matrices of the individual Models

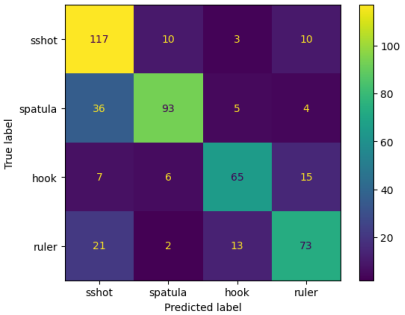
(a) Color model Action Matrix



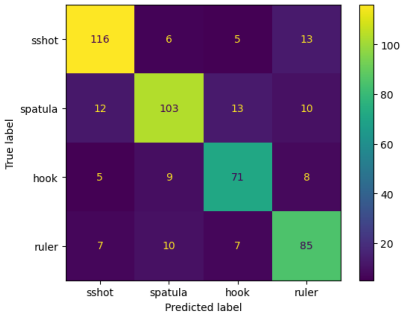
(b) Depth model Action Matrix



(c) Color model Tool Matrix



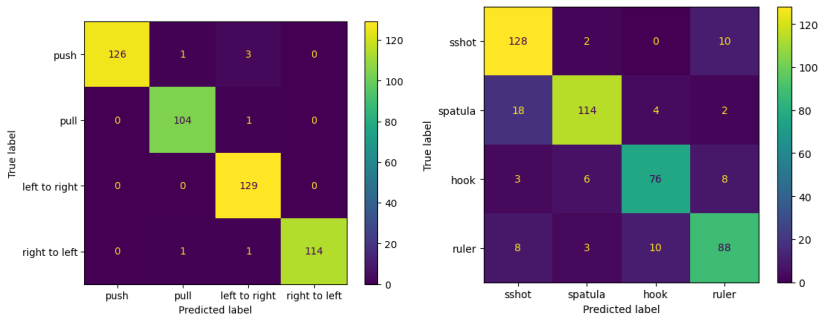
(d) Depth model Tool Matrix



0.05. These results show a notion of specialization for a given class. The color model having an advantage on actions and the depth model an advantage on tools. The precisions reflect this hypothesis. Ranging from 0.73 to 0.83, achieving approximately half of the difference between the color model tool class precisions, showing more stability in the predictions per tool class category. The matrix for the tool predictions thus reveals no clear tendency towards one category in contrast to the color model matrix. A small note would be the mislabelling of "spatula" data points, as mentioned before this fact can be traced back toward the simple action set in our dataset, simple actions may not leave concrete and distinct spatial features in the effect image for a model to discern between the used tools.

The fused model results achieve a cumulative accuracy of 0.92. The action performance reflecting the trend as seen throughout this evaluation with a near perfect accuracy. The tools increasing their accuracy from the individual models by a major 0.12 and 0.07 for the color and depth model respectively, resulting in an accuracy of 0.85. The fusing allowed for the stability of the depth model to transition to the final predictions. The difference between precisions resulting in 0.1 ranging from the highest 0.91 to 0.81. Figure 5. illustrates the matrices for the fused predictions. Two observations emerge from these matrices. The comparison between the action matrix of the fused model and the action counterparts of the individual models show a similarity between the color model and the final model. The second observation shows the opposite for the tool matrices. The depth model showing more similarity with the final model. The observations thus strengthen the hypothesis that both individual models performed

Figure 5: Confusion matrices of the action class (left) and tool class (right) for the fused model



better on the different classes given their different modalities, thus improving the final model by combining said different advantageous.

Proceeding to the comparison of the early and late fusion model. The late fusion model performed better with a difference on the overall accuracy of 0.02. The individual classes on the one hand give a nearly exact performance for action class, the early fusion model achieving better precision scores. The tool class shows the interesting diverging results and impact in scores. The aforementioned difference between the category precisions, showed stability for the late fusion model with 0.1, the early fusion model possessing a difference of 0.17, in addition to possessing two precision scores under 0.8 whereas the late fusion model precision scores perform reliably over 0.8, playing into the predictive stability of the late fusion model. The confusion matrices further increase the insight into this stability and reliability. An intriguing observation for the action class matrices is the fact that the early fusion model performed better than the late fusion model, only by three classification data points, yet it is still important to note for this comparison. In the tool class the late fusion model error counts do not exceed 20 for one given category, the second highest being 10, in contrast to the early fusion models top three error counts reaching 22, 16 and 12. The summary of the results, cumulates in the late fusion model outperforming the early fusion model, in addition to creating more reliable and stable predictions. Among other things, hinting towards a stabilizing effect in the fusion of modalities, when specialization in individual modality models occur.

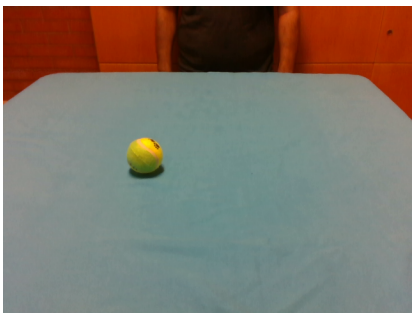
5. Discussion

The thesis started off with the main premise of comparing two specific fusion techniques with each other applied to a multi-tasking problem performed on a multimodal dataset. Through the explored examples interesting patterns, observations and results emerged. As a start, the best model decision criteria must be revisited and explained. In the methods section, the best model criteria was based on the tool accuracy and not the overall accuracy. During training as well as hyperparameter tuning, the same pattern emerged as in our final results. Tools performed worse than actions, dragging the overall accuracy down by small steps. This led to the decision to focus on the tool accuracy given the excessive performance of the action class creating no discernible distinction between test runs, in addition to the uninformative global accuracy. The

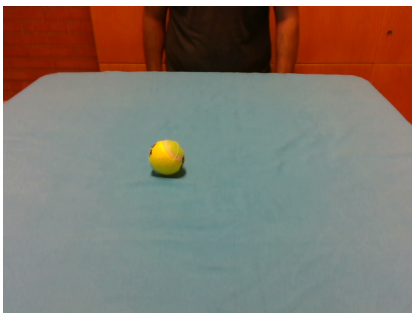
overall accuracy did not reflect the major and decisive changes of the tool class, potentially missing the key factor fusion applies to the model. Let us now dive into the early and late fusion results. The performance on the action class was near perfect, a result, which is not unsurprising for a CNN. Our actions leave a clear translation in the effect images, playing perfectly into the aforementioned spatial capturing property of CNNs and reflecting the growing interest and implementation of convolutional neural networks in computer vision, robotics and action recognition specifically (Yao, Lei, and Zhong 2019; Herath, Harandi, and Porikli 2017). A pull changes the y coordinate of an object to be higher, an observed agent pulls an object towards it, a push results in a y coordinate lower than the none effect image, pushing away from the observed agent. Tools on the other hand, do not express such an immediate, recognizable difference. The accuracy for the tool class diminishes based on this notion, many confusions appear in the matrices with no clear pattern.

Figure 6: An illustration of a datapoint, showing the lack of spatial features for classification purposes.

(a) Tennisball pushed by a Ruler



(b) Tennisball pushed by a Spatula



In Figure 6. we see two illustrations of an after action effect image of an object, the left image has been pushed by a ruler, the right image has been pushed by a spatula. The difference being subtle and mostly unnoticeable, ending in a guessing game when it comes to human observants. The model achieving an accuracy three times higher than chance level shows the underlying power of convolutional neural networks in affordance classification. A remedy for diminishing this inaccuracy in the tool class would be an expansion of our action set, using more elaborate and complex actions that leave traceable marks in spatial images. A different approach, would be the implementation of different tools, yet this would change the scope of our affordance based approach, given the inability of our models to classify all tools presented, failing the overall premise of this thesis immediately. These statements lead us to the questions looming over the experiments, inquiring about the performance of both fusion techniques individually. Early fusion was able to classify the actions and tools in a reliable manner, solving the first sub research question. The late fusion model equally performed the set tasks, reflecting the findings stating fusion being a problem depended technique and answering the second sub research question (Ramachandram and Taylor 2017). Both models performed well, given the correct time investment and hyper parameter tuning, only the comparison showed one technique being superior. The late fusion model emerging as the best between both of the implementations. As hinted towards in the result section, the late fusion model confusion matrix for tools illustrates the effect of the

individual models on the fused outcome. Both predictive advantages of the individual models were combined into one, improving the overall result. The averaging of the individual results, help to decrease uncertainty present in the individual models. An output uncertain between two different categories was averaged with another output certain about its category, which forces the uncertainty out of the final result, thus combining models with different certainties, improves the overall predictive capabilities of the fused model, given one of the fused models has lower uncertainty. This notion is reflect in the accuracy result in addition to the category precisions, as previously stated they show less deviation between individual categories and achieve a smaller difference between the lowest and highest precision. Complimenting the approach modern science applies in fusing highly optimized models on the decision level, instead of creating new models from the ground up for each and every individual new complex task. The results on the tool class show the effect of fusion and multimodal data on affordance categorization, a notion not entirely reflected in the Visual affordance survey by Hassanin, Khan, and Tahtali (2021). One out of the ten studies, explored in the affordance categorization comparison, used multi modal data and a neural network (Hassanin, Khan, and Tahtali 2021). The results therefore reinforcing the overarching foundation on multi-tasking and multimodal implementations in affordance categorization and the next step being an expansion towards affordance detection. A small note on a limitation of this paper is the exclusion of intermediate fusion. This study applies fusion either at one end or the other end of our CNN, yet the range in between is a viable option. Intermediate fusion changes the fusion level, instead of fusing on the decision layer they fuse in between layers. A great tool whenever modalities share information, which have to be fused earlier, not wasting an entire network on redundant information. This technique was excluded given the small amount of modalities and broad range of testing possibilities in fusion layer placements.

6. Conclusion

This study strived to reinforce the foundation set by multiple different branches in AI, robotics and computer vision. To that end we now answer the main research question stated in the introduction, as the sub research questions have been addressed in the discussion.

What impact do different types of multimodal fusion have on the precision and accuracy using the iCub action recognition dataset to predict action and tool duos ?

Late fusion impacts the outcome of the model by clearing out uncertainties which emerge from the individual models and combining the beneficial predictive powers of single models. In comparison the early fusion model learns an overall representation, achieving valid and good accuracies, yet with greater instability towards category precisions. Late fusion being preferred in the case of our dataset and experimental setup. The experiment showing the effectiveness of the basic tools brought by the different branches, proving their suitability for combined implementations and expandability on multi-tasking models. Future research, should tackle the issue of tool accuracy, by implementing increasingly complex and elaborate actions, to bring more diversity and discernability to the use of tools. Another expansion would be in the affordance branch, build onto our model an affordance recognition model for observed tools, presented towards the action observer, along with a robotic implementation, to test our models in an interactive setting. A final curiosity driven expansion would attempt an intermediate fusion approach on the iCub action recognition dataset and the experimental setup

presented in this study in addition to expanding the action-tool duos to object-action-tool triplets, taking the multi-tasking a step higher.

References

Baldi, Pierre and Peter J Sadowski. 2013. Understanding dropout. In *Advances in Neural Information Processing Systems*, volume 26, Curran Associates, Inc.

Bayoudh, Khaled, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa. 2021. A survey on deep multimodal learning for computer vision: Advances, trends, applications, and datasets. *The Visual Computer*, 38(8):2939–2970.

Chu, Fu-Jen, Ruinian Xu, Landan Seguin, and Patricio A. Vela. 2019. Toward affordance detection and ranking on novel objects for real-world robotic manipulation. *IEEE Robotics and Automation Letters*, 4(4):4070–4077.

Clark, Alex. 2015. Pillow (pil fork) documentation.

Falbel, Daniel. 2023. *torchvision: Models, Datasets and Transformations for Images*. Https://torchvision.mlverse.org, https://github.com/mlverse/torchvision.

Fukushima, Kunihiko. 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202.

Gibson, James J. 1979. The theory of affordances. *Hilldale, USA*, 1(2):67–82.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Hassanin, Mohammed, Salman Khan, and Murat Tahtali. 2021. Visual affordance and function understanding. *ACM Computing Surveys*, 54(3):1–35.

Herath, Samitha, Mehrtash Harandi, and Fatih Porikli. 2017. Going deeper into action recognition: A survey. *Image and Vision Computing*, 60:4–21. Regularization Techniques for High-Dimensional Data Analysis.

Hunter, J. D. 2007. Matplotlib: A 2d graphics environment. *Computing in Science and Engineering*, 9(03):90–95.

Joze, Hamid Reza Vaezi, Amirreza Shaban, Michael L. Iuzzolino, and Kazuhito Koishida. 2020. Mmtm: Multimodal transfer module for cnn fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kirtay, Murat, Ugo Albanese, Lorenzo Vannucci, Guido Schillaci, Cecilia Laschi, and Egidio Falotico. 2020. The icub multisensor datasets for robot and computer vision applications. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, ICMI '20, page 685–688, Association for Computing Machinery, New York, NY, USA.

Kniesmeijer, Kas and Murat Kirtay. 2022. icub! do you recognize what i am doing?: multimodal human action recognition on multisensory-enabled icub robot.

Lakani, Safoura Rezapour, Antonio J. Rodríguez-Sánchez, and Justus Piater. 2018. Towards affordance detection for robot manipulation using affordance for parts and parts for affordance. *Autonomous Robots*, 43(5):1155–1172.

McKinney, Wes. 2010. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*, pages 56 – 61.

Min, Huaqing, Chang'an Yi, Ronghua Luo, Jinhui Zhu, and Sheng Bi. 2016. Affordance research in developmental robotics: A survey. *IEEE Transactions on Cognitive and Developmental Systems*, 8(4):237–255.

Nguyen, Anh, Dimitrios Kanoulas, Darwin G. Caldwell, and Nikos G. Tsagarakis. 2016. Detecting object affordances with convolutional neural networks. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2765–2770.

Parker, Andrew J. 2007. Binocular depth perception and the cerebral cortex. *Nature Reviews Neuroscience*, 8(5):379–391.

Paszke, Adam, 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. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, page 12, Curran Associates, Inc.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Ramachandram, Dhanesh and Graham W. Taylor. 2017. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108.

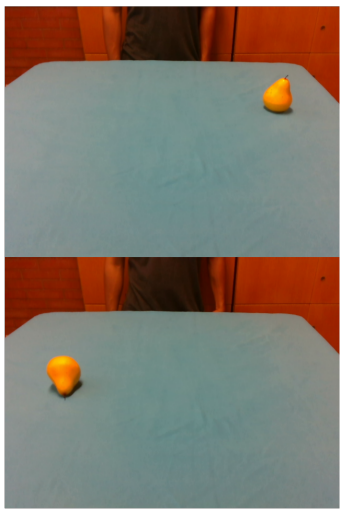
Ruder, Sebastian. 2016. An overview of gradient descent optimization algorithms. page 14.

Song, Yaoxian, Jun Wen, Dongfang Liu, and Changbin Yu. 2022. Deep robotic grasping prediction with hierarchical rgb-d fusion. *International Journal of Control, Automation and Systems*, 20(1):243–254.
Van Rossum, Guido and Fred L. Drake. 2009. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.
Van der Walt, Stefan, S. Chris Colbert, and Gael Varoquaux. 2011. The numpy array: A structure for efficient numerical computation. *Computing in Science and Engineering*, 13(2):22–30.
Yao, Guangle, Tao Lei, and Jiandan Zhong. 2019. A review of convolutional-neural-network-based action recognition. *Pattern Recognition Letters*, 118:14–22. Cooperative and Social Robots: Understanding Human Activities and Intentions.

Appendix A: Input shape

Figure 1: Orientation illustration of late fusion input matrices of a pear toy datapoint

(a) color Input (shape 64x128) (b) Depth input (shape 64x128



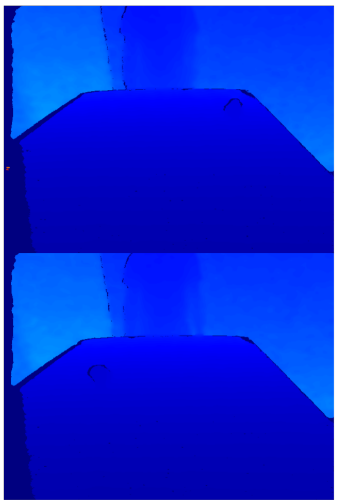


Figure 2: Orientation illustration of early fusion input matrix of a pear toy datapoint

(a) Input (shape 64x256)