Improving Robotic Grasping on Monocular Images Via Multi-Task Learning and Positional Loss

William Prew*‡, Toby Breckon*†, Magnus Bordewich*, Ulrik Beierholm‡

Department of Computer Science*, Engineering†, Psychology‡

University of Durham, UK

Abstract—In this paper we introduce two methods of improving real-time object grasping performance from monocular colour images in an end-to-end CNN architecture. The first is the addition of an auxiliary task during model training (multi-task learning). Our multi-task CNN model improves grasping performance from a baseline average of 72.04% to 78.14% on the large Jacquard grasping dataset when performing a supplementary depth reconstruction task. The second is introducing a positional loss function that emphasises loss per pixel for secondary parameters (gripper angle and width) only on points of an object where a successful grasp can take place. This increases performance from a baseline average of 72.04% to 78.92% as well as reducing the number of training epochs required. These methods can be also performed in tandem resulting in a further performance increase to 79.12%, while maintaining sufficient inference speed to afford real-time grasp processing.

I. Introduction

Robotic object grasping is a complex task, with solutions often drawing from the combined knowledge of diverse fields including physics, psychology, and computer science [1], [2].

Recently, machine learning has played a key role in simplifying the formation of grasp plans by reducing the need for hand-crafted analytical features specific to the task. For general-purpose robotics systems these hand crafted features can be difficult and time consuming to produce [2]. Machine learning offers a more hands-off data-driven approach. Deep neural networks have successfully enabled robots to perform grasping tasks autonomously, with grasp performance often exceeding that of systems with human-designed features [3], [4], and with greater generalisability for unseen objects because new objects do not require unique analytical features.

Many architectures of deep neural networks have been applied to the task of grasping. Previous works have utilised convolutional neural networks (CNN) [5], [6], [7], [8], reinforcement learning systems (RL) [9], [10], [11], and generative adversarial networks (GAN) [12] in both real settings and simulated environments [13] to form grasp plans.

Most state-of-the-art systems tend to use complex network architectures, solving a number of simultaneous tasks such as semantic relations [14] which are combined to generate a grasp proposal. These networks therefore incur significant computational overhead to the detriment of real-time performance, often taking seconds to infer grasps [15], [16]. Smaller CNN architectures, while generally slightly less accurate, tend to run a lot faster and are capable of inferring grasps in real time [17], [18].
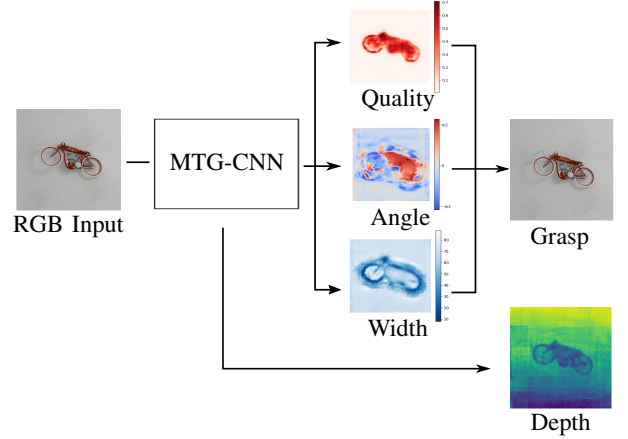

Fig. 1. An overview of the MTG-CNN approach for grasp detection and depth reconstruction.

This paper proposes two methods of improving the accuracy of a lightweight CNN for robotic grasping while retaining real time performance: 1) incorporating multi-task learning during training, and 2) using a novel positional loss function. We build on the generative grasping convolutional neural network (GG-CNN) from Morrison et al. [8], [19], a relatively small neural network which directly generates a grasp from an image. Due to the small number of parameters in the network, grasps can be generated multiple times per second for fast control of an autonomous robotic agent.

1) Multi-task learning, in which a system simultaneously learns to perform a main and an auxiliary task, has been shown to improve main task performance [20], [21] and has been increasingly applied to the field of machine learning to increase performance on a given task [22]. However, this is only seen when a network learns tasks that share a common optimal hypothesis task [23]. We observe that grasping networks given true depth as an input can outperform networks given colour monocular images, and therefore propose our multi-task grasping convolutional neural network (MTG-CNN). The MTG-CNN (see Fig. 1) network replicates the GG-CNN network but also performs an auxiliary task. When this auxiliary task is to reconstruct depth from monocular images, we obtain improved performance at grasping without adding to the complexity of network architecture in deployment and hence preserve the ability for real-time grasp operation.

2) We introduce a new positional loss function which is able to achieve faster convergence, therefore requires far fewer training examples, and achieves better grasping performance on the large Jacquard grasping dataset [24]. The intuition behind the positional loss function is that we do not penalise the network for errors on the angle or width of grasp reconstructions where the position of the grasp is anyway unsuccessful, thereby focusing the learning of the network towards task success.

II. Related Work

The task of robotic grasping is defined as the technique in which a robotic end-effector can be positioned to securely grab an object and successfully lift it [2]. The action of planning and executing a grasp can be broken down into two parts: a computer vision, and a robotics problem.

For a successful vision based grasp plan, multiple factors should be considered. The proposed grasp should be robust enough to cope between variations in the environment, such as changes to lighting or viewing angle, and the inference time of a given model should also be small enough to provide a robotics system with a grasp plan in real time. Subsequently, the robotics system must be able to decide on the correct 3-D approach vector to achieve successful gripper closure on an object [3]. Ideally, these systems should complement one another to generalise to unseen objects.

In this section we briefly outline previous literature that relates to the formation of a successful grasp plan:

A. Grasp Synthesis

In the context of robotics, grasp synthesis refers to the problem of finding a grasp configuration that satisfies a set of criteria relevant for a grasping task [3]. According to Sahbani and colleagues [25], previous solutions can be broadly categorised into either analytical and empirical techniques.

Analytical techniques require the use of hand-crafted features, typically generated by humans, to generate application specific parameters. These hand-crafted features can include: specific geometric features, or mathematical or physical models to calculate stable grasps [3]. However, these are typically generated on the premise that these properties are known. Therefore it is difficult to generate such features for every object leading to problems with the generalisation of such techniques [26].

Empirical (data-driven) techniques on the other hand, are developed using existing knowledge of object grasping and rely on previously known successful results [3]. One example of these has been the more recently applied deep learning models which have been used to generate grasp points via learned features such as grasp affordances [14] or semantic labels [27]. Machine learning models are now often used to form effective grasp plans for robots [16], [17] including in real world settings e.g. [28].

These empirical approaches are now the generally preferred method over their analytical counterparts due to their ability to generalise better to unseen objects [2] and their superior

inference time allows for the real-time control of robotic agents [8], [18], [29]. Some models have even attempted to improve overall execution time by pre-processing grasp candidates [15] or trading off execution time for the number of sampled grasps [4].

The main limitation of such empirical methods are that they require a large amount of detailed data in order to infer grasps [17]. Some studies have attempted to improve generalisation to new environments with potentially far less available data with the introduction of techniques such as online learning using pretrained models [30] or from transferring from simulation data to real world settings [13]. However, these models tend to rely on highly detailed point cloud data [31] or colour and depth (RGB-D) input [6], [29] to train models which is not always available to certain systems. There is a section of research that attempts to improve results on impoverished colour only data by joint training on related tasks, inspired by mammalian models of vision [32].

Here we build on the empirical work of Morrison et al. [8] by using a relatively small CNN. We are able to produce grasps with fast inference times by inferring grasp plans for every pixel simultaneously on a typical colour or depth image. The output from such models can then be passed to a robotic system for online, real-time control [19].

B. Grasping Datasets

With the advent of data-driven grasping, new datasets have subsequently been proposed to provide large scale data reserves for training models. One of the earliest datasets was the popular and widely used Cornell grasping dataset (CGD) [33]. This contained an initially limited 885 images of 240 different object types, and a total of 8019 valid and invalid hand-labelled grasps. It was designed for common parallel plate grippers and was used extensively as a benchmark c.f. [6], [34], [18]. As each grasp was labelled by a human annotator, Mahler [16] notes multiple drawbacks to the process, as it was tedious to collect all available grasps and therefore difficult to generalise the hand-labelling process to other objects or scenes. Furthermore, this process likely introduced human bias into the robotic grasping data as the annotator is likely to label a grasp that is easier for humans to pick up but not necessarily for a parallel plate gripper attached to a robot arm [4], [24].

Since the CGD, more contemporary datasets have been developed that aim to facilitate the formation of grasp synthesis plans on different robotic arms. Thereby, improving their ability to generalise to different environments. Some datasets have focused on generating a larger amount of data for empirical methods to train from [4], [24], or developing an open-source platform to consolidate data from multiple robotic arms [35]. Other datasets have focused on improving the replicability of previously developed models between different arms by producing datasets filled with standardised items [36], [37], or developing more detailed datasets for grasping on robot arms with access to a greater number of degrees of freedom [27], [38].
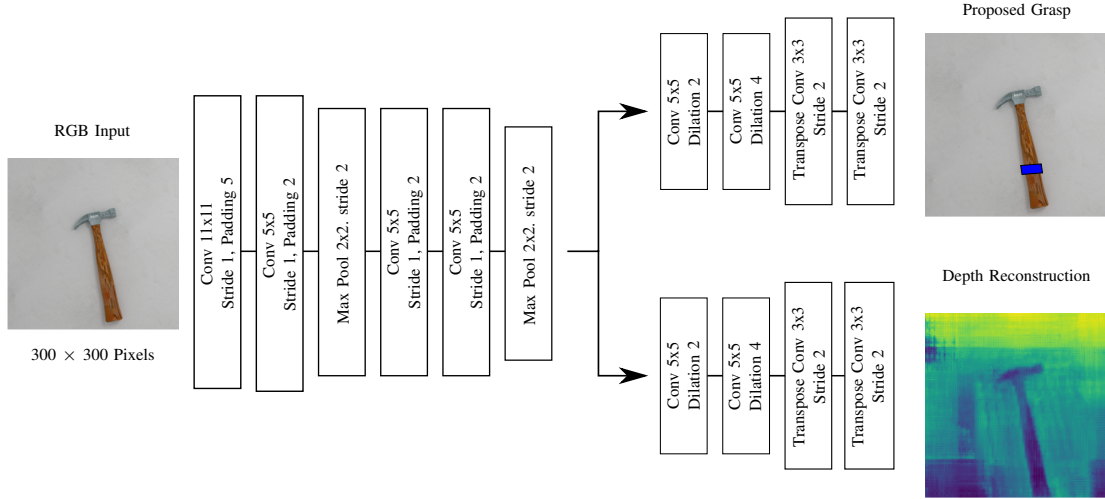
Fig. 2. Architecture of the MTG-CNN network. The GG-CNN architecture follows the same structure except with the auxiliary task branch omitted.

Most of these datasets have the common benefit over the CGD in that they have generated a larger amount of labelled grasps to train data-driven models. Another commonality is that most of these datasets are typically now labelled by the robot arms themselves compared to the hand-labelled data from the CGD for more accurate training data. This has been achieved through procedures such as autonomous collection from real robots [17], or by generating a large number of grasping examples using simulated data [13]. This is done in both an online fashion through reinforcement learning [36], or by using a simulated robot arm to attempt a larger number of grasps like the more recent Jacquard grasping dataset [24] that was used to train and evaluate the model presented here.

This is a large-scale grasping dataset where all grasps are developed and labelled in a simulated environment. It contains over 1.1 million grasps for over 11,000 objects which is a widely used benchmark amongst data-driven antipodal grasping models c.f. [8], [29].

C. Multi-Task Learning

Multi-task learning (MTL) is a machine learning approach that aims to help models better generalise to a given main task by using the training signals of a related task to learn an appropriate inductive bias [20]. MTL has been shown to improve performance on a main task when simultaneously training on a simpler auxiliary task across a wide range of disciplines [22]. By sharing features between multiple tasks, the network is forced to learn common representations between them that may reduce overfitting, resulting in better generalisation to the original task. Examples include computer vision (e.g. using the characteristics of the road to predict steering direction [20] or the segmentation of images for classification in the fast R-CNN model[39]); natural language processing [40], [41]; and speech recognition [42].

MTL can be implemented through an assortment of different functions, either separately or together in the same model; as long as the model has access to all the common representations required for both tasks. Such examples can include the cross-talk of information between different networks [41], focusing attention on a given task by learning an auxilliary training bias [20], or leveraging more specific examples like representation learning for language modelling [43]. Some studies do state for MTL to be effective that related tasks need to share a similar optimal inductive bias [21]. As such, it is not always clear which auxiliary tasks will lead to performance increases on the main task during joint learning [22]. Some studies have attempted to bypass this problem with specifically designed multi-task losses that add an extra coefficient for training a plethora of different techniques [44].

Most current deep learning implementations for robotic grasping tend to only focus on a single grasping task [5], [17], [18]. However, specific grasping examples have shown that there is a relative performance increase when networks are trained while performing auxiliary tasks, such as semantic segmentation [45], [46], [32], [27], [14]; saliency detection [47]; or the generation of bounding boxes [48]. Such MTL techniques can be applied to the model using either a bottom-up [47], [14] or a top-down approach [32], [27]. Similarly, in the current study we attempt to improve overall grasping performance by forcing shared parts of the network to learn a similar auxilliary task, when presented with only monocular colour data.

III. METHODOLOGY

In this section we give a formal definition of the grasping problem we address, then present the network architecture and loss functions used in our solution and finally describe the dataset used for training and evaluation.

A. Grasping Problem

The grasping problem we address is to take a colour (RGB) input image I of an object and output a proposed grasp for a robotic arm with an antipodal gripper approaching perpendicularly to the x-y plane. The proposed grasp is given as a tuple $((x, y), \Phi, W)$, where (x, y) is the central position of the gripper in the x-y plane relative to the image, Φ is the angle of rotation of the gripper around the z-axis, and W is the width of the gripper jaws for the grasp attempt.

A proposed grasp corresponds to real world coordinates through a series of known transforms in Cartesian coordinates via the below equation:

$$((x, y), \Phi, W) = t_{RC}(t_{CI}((x, y), \Phi, W)) \tag{1}$$

where t_{CI} is the transform from the 2D image coordinates into the 3D camera frame, and t_{RC} is the transform from the camera frame to the world or robot frame with known intrinsic parameters and known calibration between the camera and the robot.

B. Multi-task Network Architecture

Our proposed solution is a multi-task grasping convolutional neural network (MTG-CNN), based on the generative grasping convolutional neural network (GG-CNN) [8], [19]. The GG-CNN is a small and lightweight network for closed-loop continuous grasping to produce a map of potential antipodal grasps perpendicular to the x-y plane. Our experiments use the GG-CNN model as defined in [19] as the backbone. The MTG-CNN network extends the GG-CNN network by branching at the halfway point of the GG-CNN network and duplicating the second half. One branch is used for learning the grasp outputs, as in GG-CNN, the other branch for an auxiliary learning task, shown as depth reconstruction in Fig. 2. The shared first half consists of two convolutional layers, followed by a max pooling layer and two further convolutional layers and a second max pooling layer. The two branches in the second half of the network are identical and consist of two dilation layers and two convolutional transpose layers, see Fig. 2. All convolutional layers have 16 filters except for the dilated convolutional layers which have 32 filters.

The MTG-CNN network can be viewed as a function M_θ, where θ represents the model weights, which maps a 300×300 pixel input image I to five parameter-space output images $Q_\theta(I), \Phi_\theta^{\sin}(I), \Phi_\theta^{\cos}(I), W_\theta(I)$ and $A_\theta(I)$. The first four of these form the grasping output and come from one branch of the network, and the final auxiliary output A_θ comes from the the other, auxiliary, branch. The pixel values of $Q_\theta(I)$ represent the probability of a successful grasp at each pixel position in I. The angle of the proposed grasp can be calculated pixelwise by $\Phi_\theta(I) = \arctan(\frac{\Phi_\theta^{\sin}(I)}{\Phi_\theta^{\cos}(I)})/2$. The raw angle Φ of each grasp is in the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$; it is represented as two unit vector components $\cos(2\Phi)$ and $\sin(2\Phi)$ in the network outputs as this aids learning by removing the discontinuity at the wrap-around point [49]. Finally, $W_\theta(I)$ is the required gripper width as a value in the range $[0, 1]$, which is scaled
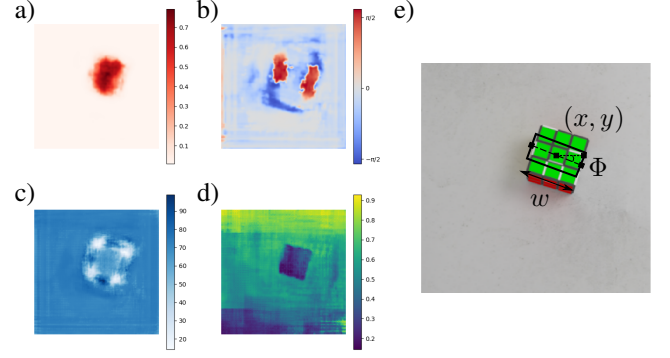

Fig. 3. Example output from the MTG-CNN model that contributes towards a labelled grasp image. a) The probability of grasp success Q for each pixel in the range of $[0, 1]$, b) The proposed grasp angle Φ at each pixel generated during post-processing from $\Phi_{\sin}$ and $\Phi_{\cos}$ in the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$. c) The required width of the gripper W represented pixel-wise in the range $[0, 150]$. d) The generated depth map image from an RGB input normalised between the values $[0, 1]$. This does not contribute to the final grasp. e) An example grasp. The pixel with the highest grasp probability in Q is used as the centre of the grasping rectangle. The angle and width are calculated from the same pixel coordinates (x, y) from each image.

into a physical measurement for gripper width in the range of $[0, 150]$. Fig. 3 illustrates an example input and outputs of MTG-CNN.

The GG-CNN network, which does not feature the auxiliary branch, is similar but produces only the four parameter space output images $Q_\theta(I), \Phi_\theta^{\sin}(I), \Phi_\theta^{\cos}(I)$ and $W_\theta(I)$ corresponding to the grasping outputs from the original branch of the MTG-CNN network.

To extract a proposed grasp from either network we take the x, y coordinates of the maximum pixel value in $Q_\theta(I)$ as the centre of the grasp, and then extract the corresponding pixel values from $\Phi_\theta(I)$ and $W_\theta(I)$ to obtain the angle and width of the gripper. For smoothness, before extracting the proposed grasp the outputs $Q_\theta(I), W_\theta(I)$ and $\Phi_\theta(I)$ are filtered with a Gaussian kernel and a standard deviation of 5 pixels, as is done in [50].

At inference (test) time only the parameterized output of the grasp is required, so the additional complexity of the auxiliary branch in MTG-CNN can be discarded resulting in streamlined real-time capable performance.

The purpose of multi-task learning is to guide the shared layers of the network towards learning features that will be useful for the grasping task, which it might not otherwise find. Preliminary experiments on the GG-CNN network given true depth images as input, instead of colour images, showed improved performance at grasping. This motivated us to use depth reconstruction as an auxiliary task, with the intuition that the MTG-CNN network would learn features related to 3D structure in the shared layers, and these would be useful for the grasping task. In addition to depth reconstruction, we also tested saliency detection as an auxiliary task, with the intuition that the network needs to identify the object in the scene, and features tuned towards this task in the shared layers may be helpful.

C. Positional Loss

This paper introduces a more focused loss function, which we call *positional loss*, for use during training. The loss function used with the GG-CNN network in [19] is the summation of the MSE loss of each of the four parameter space output images $Q_\theta(I), \Phi_\theta^{\sin}(I), \Phi_\theta^{\cos}(I), W_\theta(I)$. This can be extended to a loss function L for the MTG-CNN network by simply including the MSE of the auxiliary output in addition:

$$
\begin{aligned}
L = & \frac{1}{N} \sum (Q_\theta - Q_{GT})^2 \\
& + \frac{1}{N} \sum (\Phi_\theta^{\cos} - \Phi_{GT}^{\cos})^2 \\
& + \frac{1}{N} \sum (\Phi_\theta^{\sin} - \Phi_{GT}^{\sin})^2 \\
& + \frac{1}{N} \sum (W_\theta - W_{GT})^2 \\
& + \frac{1}{N} \sum (A_\theta - A_{GT})^2
\end{aligned}
\tag{2}
$$

where N is the number of pixels per image, the summations are over the pixels of each image, θ denotes the output of the network and GT denotes the ground truth from the dataset.

The loss function L includes a significant contribution from the error in angle prediction and width prediction even at positions of the input image where no successful grasp is possible, for example: regions containing the background not the object. The network must learn to output values close to a default value of 0 for angle and width in such regions even though no grasp will be attempted at these positions and therefore these angle and width parameters are not used. The positional loss function L_p scales the angle and width errors by ground truth for the quality of the grasp (Q_{GT}) at each position in the scene. The scaling of the loss in this manner focuses the learning of the network on grasp quality and on angles and widths only at positions where grasp attempts may occur. The positional loss function is defined below:

$$
\begin{aligned}
L_p = & \frac{1}{N} \sum (Q_\theta - Q_{GT})^2 \\
& + \frac{1}{N} \sum (Q_{GT}(\Phi_\theta^{\cos} - \Phi_{GT}^{\cos}))^2 \\
& + \frac{1}{N} \sum (Q_{GT}(\Phi_\theta^{\sin} - \Phi_{GT}^{\sin}))^2 \\
& + \frac{1}{N} \sum (Q_{GT}(W_\theta - W_{GT}))^2 \\
& + \frac{1}{N} \sum (A_\theta - A_{GT})^2 .
\end{aligned}
\tag{3}
$$

For the GG-CNN model, the loss functions L and L_p are as above with the final summation, with the extra auxiliary term removed. Networks that are trained with the new positional loss function are marked with a subscript p.

D. Dataset

All models are trained with the Jacquard grasping dataset [24]. This is a simulated dataset containing 54,485 images of over 11,000 different objects on uniform white backgrounds. The images are annotated with over 1.1 million successful
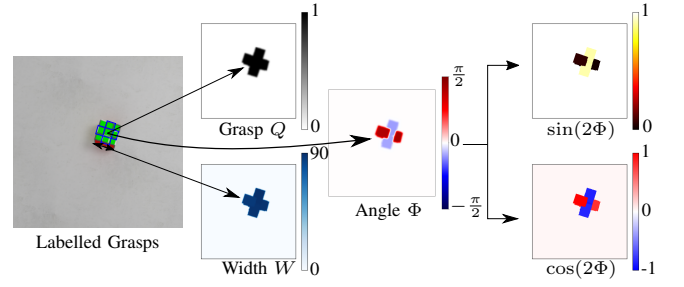

Fig. 4. Annotated grasps from the dataset are transformed into images for training. Grasps are broken down into grasp quality Q, grasp angle Φ, and grasp width W.

grasps represented as grasp centre, angle and gripper width. All grasps were generated within a simulated physics environment which attempted grasps at many positions, angles and widths; unsuccessful grasps and highly similar grasps were not recorded.

Every object in the dataset has four viewing angles and each viewpoint consists of a single RGB image as well as a perfect depth image recorded from the simulated data and a generated stereo depth image. The perfect depth image was used to train the depth reconstruction branch.

For each image, I, ground truth parameter-space images, $Q_{GT}(I), \Phi_{GT}^{\sin}(I), \Phi_{GT}^{\cos}(I), W_{GT}(I)$ and $A_{GT}(I)$, are computed as follows (see Fig. 4). For $Q_{GT}(I)$ the pixel values are 1 if the pixel falls within the centre third of the successful grasping rectangle for I and 0 otherwise. Values of $\Phi_{GT}^{\cos}(I), \Phi_{GT}^{\sin}(I)$ and $W_{GT}(I)$ are set to according to the angle and width of a corresponding successful grasp centred at that pixel position, where $Q_{GT}(I) = 1$ and are set to 0 elsewhere. The angle Φ of the grasp is in the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$, and is decomposed into two unit vector components $\cos(2\Phi)$ and $\sin(2\Phi)$, which are used for $\Phi_{GT}^{\cos}(I), \Phi_{GT}^{\sin}(I)$. When the auxiliary branch is learning depth reconstruction, $A_{GT}(I)$ is set to the true depth data of the simulated scenes normalised to the range $[0, 1]$. When the auxiliary branch is learning saliency, $A_{GT}(I)$ is set to the value of a binary mask provided with the dataset, where 1 indicates a pixel displaying part of the object, and 0 a pixel displaying the background.

For training and testing, the dataset is split in a similar manner to [24]: a 5-fold cross validation was performed in which the 80% of the data was reserved for training, and the remaining 20% of the data was split evenly between a validation and test set. Colour input data was normalised to the range of $[0, 1]$ before subtracting the image mean to centre the data around 0. All models were trained for 40 epochs which preliminary testing showed was sufficient for full training. Furthermore, data augmentation was applied to the dataset to artificially increase the amount of training data by random cropping and zooming on the images.

Performance on the dataset is measured using the intersection-over-union (IoU): a grasp is considered successful if the predicted grasp rectangle shares a greater than 25% IoU and aligns within $30°$ with the ground truth grasping
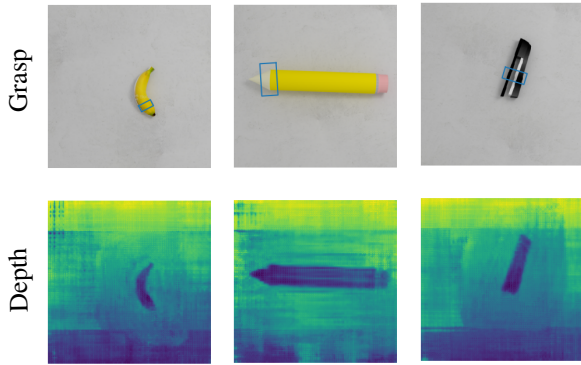
Fig. 5. Example grasp candidates and depth reconstruction from the MTG-CNN network. We suspect that previous compression of images has caused a boxing artefact that results in a grid-like structure during depth reconstruction.

TABLE I
THE MEAN PERCENTAGE OF CORRECT GRASPS FOR EACH NETWORK
EVALUATED ON THE JACQUARD GRASPING DATASET.

Model	Auxiliary Task	Successful Grasps
GG-CNN		$72.04\% \pm 3.44$
GG-CNN$_p$		$78.92\% \pm 0.97$
MTG-CNN	Saliency	$74.93\% \pm 1.86$
MTG-CNN$_p$	Saliency	$76.23\% \pm 2.75$
MTG-CNN	Depth	$78.14\% \pm 0.65$
MTG-CNN$_p$	**Depth**	**$79.12\% \pm 1.40$**

rectangle. This measure provides an efficient offline metric for testing performance and was used in both the Jacquard [24] and Cornell [33] datasets, the values here are from the Jacquard dataset [24].

IV. EVALUATION

A 5-fold cross validation was used to evaluate all the models. After training was completed, each network was tested on every object in the test set, and the percentage of successful grasp proposals under the IoU metric recorded (see Section III-D). The average number of correct grasps and standard deviation of these across the five test folds is shown in Table I.

These results show that by training on either auxiliary task we tested, the MTG-CNN network performs better than the base GG-CNN model. With the auxiliary saliency detection task, the grasping performance increases from the average baseline of 72.04% to 74.93%. With the auxiliary depth reconstruction task we see a larger increase, an average performance of 78.14%, and the network becomes more consistent overall. This implies that by training a deep neural network to perform simultaneous depth reconstruction, we can improve the results for the primary grasping task by forcing the early shared part of the network to learn key structural information about the object. We include some example outputs of the MTG-CNN network trained on depth in Fig. 5.

It can also be seen that training with the positional loss function identifies a greater number of correct grasps compared to training with the previous GG-CNN loss function: grasping performance increases from an average of 72.04% to 78.92% on the GG-CNN network, and also shows an increase for the MTG-CNN network with either auxiliary function.

When both an auxiliary task and the positional loss function are combined, we obtain the best results. The best performing network is the model trained on the auxiliary depth reconstruction task with 79.12%. At first glance these results seem show that the positional loss function only provides a small additional benefit to performance for the MTG-CNN architecture, in particular when the auxiliary function is depth reconstruction. However, by looking at Fig. 6 we see that all networks trained on the new positional loss function perform better on average that the previous loss function. During the initial parts of training, the positional loss function is achieving higher validation performance compared to the previous loss function. Therefore, the positional loss networks are actually learning the grasping task at a quicker rate, as well as attaining better performance with fewer training examples.

The GG-CCN model has 70,000 parameters, whereas the MTG-CNN model has 115,000 parameters, due to the additional branch. This slightly increases the training time per epoch, but the network remains significantly smaller than typical deep learning grasping networks. Furthermore, we do not increase inference times significantly. On a desktop NVIDIA GTX 1080Ti GPU-accelerated PC we achieve inference times between 10-25ms for the base GG-CNN model and 15-30ms for the dual-task MTG-CNN model, thus both models operate at over 30FPS. Moreover, during deployment the auxiliary task does not contribute to the grasp formation and therefore the auxiliary branch of the network can be removed. This results in identical inference times for MTG-CNN and GG-CNN, at at an average of over 50FPS.

It should also be noted that we chose the depth reconstruction as it has been shown in previous work that depth information provides key structural object information required for grasping [8]. It was believed that this task would improve the performance on monocular data according to studies like He et al. [51] and Raffel et al. [52] that show similar tasks achieve greater performance which is the reasoning behind using MSE loss for both grasping and auxiliary tasks. However, by analysing our depth reconstructions from the model, this type of loss function appears to produce artefacts that we believe to be as a result of image compression during data collection.

In future, performance could be improved with tailored loss functions other than typical MSE loss. Depth specific reconstruction losses could be used that achieve a greater smoothing effect such as that of [53] or adding the use of specific multi-task loss functions like [44]. Alternatively, tasks that are not directly related to grasping like semantic understanding [27] or grasp affordances [14] may also be applicable and explored in follow up work.
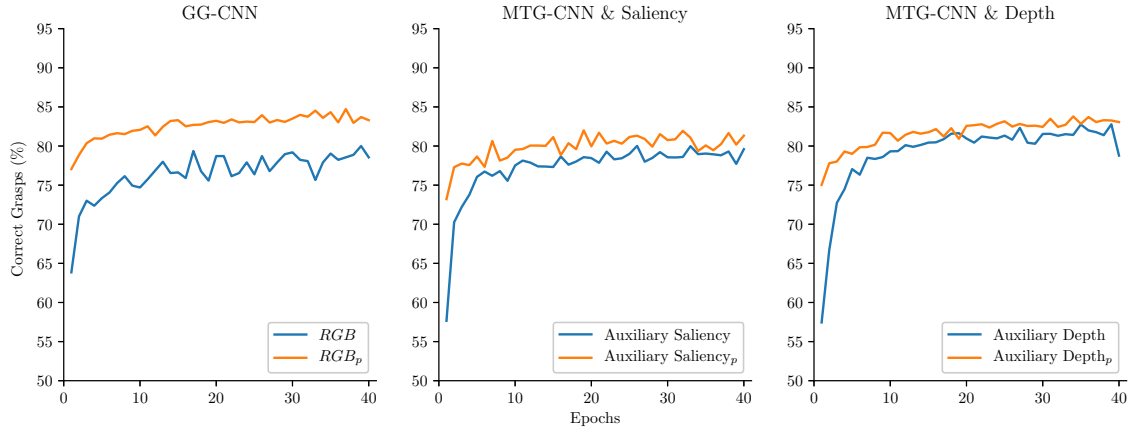
Fig. 6. Average grasping performance of each model on the validation set after each epoch. Models trained with the new positional loss function maintain faster convergence than those trained with just the previous grasping loss.

V. CONCLUSION

The first aim of this paper was to investigate whether multi-task learning can improve grasping performance when trained on auxiliary tasks with a relationship to grasping. By performing concurrent depth reconstruction with grasping, we significantly improve grasp performance on a lightweight network with only a small number of trainable parameters, and retain the speed of grasp inference.

Secondly, by introducing a new positional loss function we can train a model that converges faster and attains a further increase in grasp accuracy compared to the previous implemented loss function. The positional loss function removes learning of unnecessary output parameters (angle and gripper width) for output values of low grasp quality which do affect final grasp output, thereby allowing the network to focus learning on parameters affecting task performance itself.

REFERENCES

[1] P. Gaussier and A. Pitti, "Reaching and Grasping : what we can learn from psychology and robotics," *HAL*, pp. 1–11, 2017.

[2] S. Caldera, A. Rassau, and D. Chai, "Review of Deep Learning Methods in Robotic Grasp Detection," *Multimodal Technologies and Interaction*, vol. 2, no. 3, pp. 1–24, sep 2018.

[3] J. Bohg, A. Morales, T. Asfour, and D. Kragić, "Data-driven grasp synthesis-A survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2014.

[4] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, sep 2016, pp. 3406–3413. [Online]. Available: http://arxiv.org/abs/1509.06825

[5] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, 2015, pp. 1316–1322.

[6] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe, pp. 769–776, 2017.

[7] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, jan 2019.

[8] D. Morrison, P. Corke, and J. Leitner, "Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach," pp. 1–10, 2018. [Online]. Available: http://arxiv.org/abs/1804.05172

[9] S. Amarjyoti, "Deep Reinforcement Learning for Robotic Manipulation-The state of the art," pp. 1–18, 2017. [Online]. Available: http://arxiv.org/abs/1701.08878

[10] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning Synergies between Pushing and Grasping with Self-Supervised Deep Reinforcement Learning," *IEEE International Conference on Intelligent Robots and Systems*, pp. 4238–4245, 2018.

[11] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, "QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation," in *Conference on Robotic Learning*, no. CoRL, 2018, pp. 1–23. [Online]. Available: http://arxiv.org/abs/1806.10293

[12] K. Rao, C. Harris, A. Irpan, S. Levine, J. Ibarz, and M. Khansari, "RL-CycleGAN: Reinforcement Learning Aware Simulation-To-Real," in *Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 157–11 166. [Online]. Available: http://arxiv.org/abs/2006.09001

[13] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-to-Real via Sim-to-Sim: Data-efficient Robotic Grasping via Randomized-to-Canonical Adaptation Networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. [Online]. Available: http://arxiv.org/abs/1812.07252https://arxiv.org/pdf/1812.07252.pdf

[14] P. Ardon, E. Pairet, R. P. Petrick, S. Ramamoorthy, and K. S. Lohan, "Learning Grasp Affordance Reasoning through Semantic Relations," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4571–4578, jun 2019. [Online]. Available: http://arxiv.org/abs/1906.09836

[15] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[16] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-Net 2.0: Deep learning to plan Robust grasps with synthetic point clouds and analytic grasp metrics," *Robotics: Science and Systems*, vol. 13, mar 2017. [Online]. Available: http://arxiv.org/abs/1703.09312

[17] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018. [Online]. Available: http://arxiv.org/abs/1603.02199

[18] U. Asif, J. Tang, and S. Harrer, "Densely Supervised Grasp Detector (DSGD)," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8085–8093, 2019. [Online]. Available: www.aaai.org

[19] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 183–201, mar 2020.

[20] R. Caruana, "Multitask Learning," *Autonomous Agents and Multi-AGent SYstems*, vol. 27, no. 1, pp. 95–113, 1998. [Online]. Available: http://link.springer.com/10.1007/3-540-49430-8{_}9

[21] J. Baxter, "A Model of Inductive Bias Learning," *Journal of Artificial Intelligence Research*, vol. 12, p. 149, 2000.

[22] S. Ruder, "An Overview of Multi-Task Learning in Deep Neural Networks," *ArXiv*, jun 2017. [Online]. Available: http://arxiv.org/abs/1706.05098

[23] S. Ben-David and R. Schuller, "Exploiting task relatedness for multiple task learning," in *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, vol. 2777, 2003, pp. 567–580.

[24] A. Depierre, E. Dellandrea, and L. Chen, "Jacquard: A Large Scale Dataset for Robotic Grasp Detection," in *IEEE International Conference on Intelligent Robots and Systems*, 2018, pp. 3511–3516.

[25] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3D object grasp synthesis algorithms," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326–336, mar 2012.

[26] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," *Proceedings-IEEE International Conference on Robotics and Automation*, vol. 1, pp. 348–353, 2000.

[27] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes," in *Robotics: Science and Systems (RSS)*, nov 2018. [Online]. Available: http://arxiv.org/abs/1711.00199

[28] C. Zhihong, Z. Hebin, W. Yanbo, L. Binyan, and L. Yu, "A vision-based robotic grasping system using deep learning for garbage sorting," *Chinese Control Conference, CCC*, pp. 11223–11226, 2017.

[29] S. Kumra, S. Joshi, and F. Sahin, "Antipodal Robotic Grasping using Generative Residual Convolutional Neural Network," p. 8, sep 2020. [Online]. Available: http://arxiv.org/abs/1909.04810

[30] R. Julian, B. Swanson, G. S. Sukhatme, S. Levine, C. Finn, and K. Hausman, "Efficient Adaptation for End-to-End Vision-Based Robotic Manipulation," pp. 1–13, apr 2020. [Online]. Available: http://arxiv.org/abs/2004.10190

[31] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-Net 3.0: Computing Robust Vacuum Suction Grasp Targets in Point Clouds Using a New Analytic Model and Deep Learning," in *Proceedings - IEEE International Conference on Robotics and Automation*, sep 2018, pp. 5620–5627. [Online]. Available: http://arxiv.org/abs/1709.06670

[32] E. Jang, S. Vijayanarasimhan, P. Pastor, J. Ibarz, and S. Levine, "End-to-End Learning of Semantic Grasping," in *Conference on Robotic Learning*, 2017, pp. 1–14. [Online]. Available: http://arxiv.org/abs/1707.01932

[33] A. Saxena, "Cornell Grasping Dataset." [Online]. Available: pr.cs.cornell.edu/grasping

[34] D. Guo, F. Sun, H. Liu, T. Kong, B. Fang, and N. Xi, "A hybrid deep architecture for robotic grasp detection," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1609–1614, 2017.

[35] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn, "RoboNet: Large-Scale Multi-Robot Learning," *ArXiv*, no. CoRL, pp. 1–13, oct 2019. [Online]. Available: http://arxiv.org/abs/1910.11215

[36] B. Yang, D. Jayaraman, J. Zhang, and S. Levine, "REPLAB: A reproducible low-cost arm benchmark for robotic learning," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, may 2019, pp. 8691–8697. [Online]. Available: http://arxiv.org/abs/1905.07447

[37] D. Morrison, P. Corke, and J. Leitner, "EGAD! an Evolved Grasping Analysis Dataset for diversity and reproducibility in robotic manipulation," *arXiv e-prints*, p. arXiv:2003.01314, Mar. 2020.

[38] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "GraspNet-1Billion : A Large-Scale Benchmark for General Object Grasping," in *CVPR*, 2020. [Online]. Available: www.graspnet.net.

[39] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, 2015, pp. 1440–1448. [Online]. Available: https://github.com/rbgirshick/

[40] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 160–167. [Online]. Available: http://wordnet.princeton.edu

[41] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," in *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, jan 2020, pp. 4487–4496. [Online]. Available: http://arxiv.org/abs/1901.11504

[42] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, oct 2013, pp. 8599–8603.

[43] M. Rei, "Semi-supervised multitask learning for sequence labeling," in *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, vol. 1, apr 2017, pp. 2121–2130. [Online]. Available: http://arxiv.org/abs/1704.07156

[44] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 7482–7491, 2018.

[45] D. Katz, A. Venkatraman, M. Kazemi, J. A. Bagnell, and A. Stentz, "Perceiving, learning, and exploiting object affordances for autonomous pile manipulation," *Autonomous Robots*, vol. 37, no. 4, pp. 369–382, 2014. [Online]. Available: http://www.youtube.com/user/pileRSS2013/videos.

[46] H. Dang and P. K. Allen, "Semantic grasping: Planning task-specific stable robotic grasps," *Autonomous Robots*, vol. 37, no. 3, pp. 301–316, 2014.

[47] D. A. Klein, B. Illing, B. Gaspers, D. Schulz, and A. B. Cremers, "Hierarchical Salient Object Detection for Assisted Grasping," in *International Conference on Robotics and Automation*, 2017, pp. 1–15.

[48] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Object-based affordances detection with Convolutional Neural Networks and dense Conditional Random Fields," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe, no. September. Institute of Electrical and Electronics Engineers Inc., dec 2017, pp. 5908–5915.

[49] K. Hara, R. Vemulapalli, and R. Chellappa, "Designing Deep Convolutional Neural Networks for Continuous Object Orientation Estimation," pp. 1–10, feb 2017. [Online]. Available: http://arxiv.org/abs/1702.01499

[50] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem, pp. 4461–4468, 2016.

[51] K. He, R. Girshick, and P. Dollar, "Rethinking imageNet pre-training," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-Octob, no. ii, pp. 4917–4926, 2019.

[52] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," oct 2019. [Online]. Available: http://arxiv.org/abs/1910.10683

[53] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised Monocular Depth Estimation with Left-Right Consistency," 2017.