Evaluation of Deep Learning Techniques on a Novel Hierarchical Surgical Tool Dataset

Mark Rodrigues[1], Michael Mayo[1], and Panos Patros[2]

[1] Department of Computer Science, University of Waikato, Hamilton, New Zealand
[2] Department of Software Engineering, University of Waikato, Hamilton, New Zealand

Abstract. A new hierarchically organised dataset for artificial intelligence and machine learning research is presented, focusing on intelligent management of surgical tools. In addition to 360 surgical tool classes, we create a four level hierarchical structure for our dataset defined by 2 specialities, 12 packs and 35 sets. We employ different convolutional neural network training strategies to evaluate image classification and retrieval performance on this dataset, including the utilisation of prior information in the form of a taxonomic hierarchy tree structure. We evaluate the effects of image size and the number of images per class on model predictive performance. Experiments with the mapping of image features and class embeddings in semantic space using measures of semantic similarity between classes show that providing prior information results in a significant improvement in image retrieval performance on our dataset.

Keywords: Surgical tool dataset · semantic similarity · hierarchy tree · surgery hierarchy

1 Introduction

Surgical tool management in hospitals is a difficult, time consuming and costly task; lost, misplaced or unavailable surgical tools were estimated to cost just one New Zealand hospital over NZ$500,000 annually (Unit Manager, personal communication, Nov. 2019). Challenges faced in management of these tools included high inventory levels, multiple surgical tool set assembly errors, high staffing requirements, high costs, inconsistent availability of surgical tools, and non-functional or broken instruments being presented at surgery. Large volumes and varieties of surgical tools (Fig. 1) also pose a formidable challenge for management. According to Stockert and Langerman [21], just one institution can process over 100,000 surgical trays and 2.6 million tools every year. With an average of 38 surgical instruments present per tray, and six trays deployed on average per surgery [14], managing this volume and complexity manually under mission-critical conditions is a challenging task. Surgical tool detection and recognition through artificial intelligence (AI) and machine learning systems can provide a solution that can reduce incidents of lost or misplaced tools, improve packing

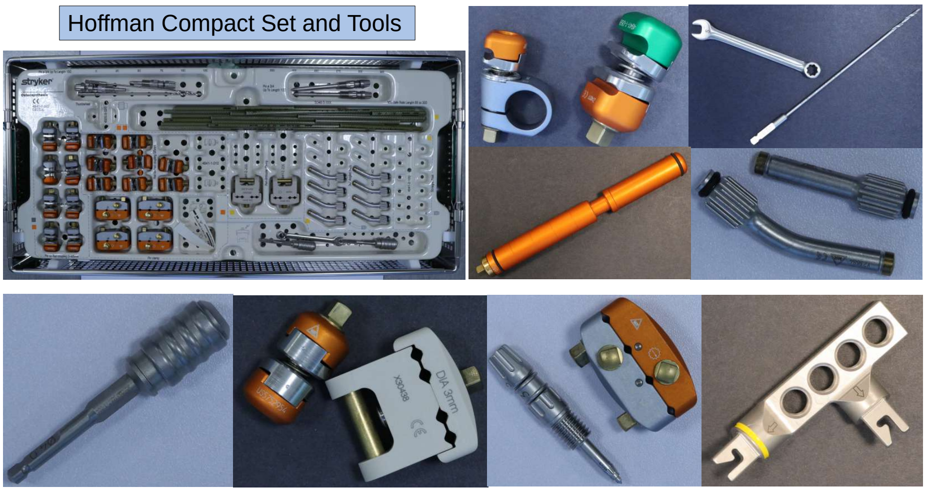

Fig. 1. Surgical Set and Tool Examples

accuracy, reduce errors, lower costs, and improve overall efficiencies within hospitals. Surgical tool recognition can be used in AI based hospital inventory management systems, and also in robotic and computer-assisted surgery, instrument position recognition, and in surgical monitoring, audit and training [11, 18, 24].

Table 1. Current Tool Datasets

Characteristic	CATARACTS [2]	Cholec80 [22]	EndoVis2017 [3]	ROBUST-MIS19 [16]
Size	50 videos	80 Videos	10 Videos	30 Videos
Focus	Cataract Surgeries	Cholecystectomy Surgeries	Abdominal (Porcine)	Varied Surgeries
Use Case	Detection	Detection	Segmentation	Detection
Classes	21	7	7	2
Annotations	Binary	Bounding Boxes	Masks	Masks
Structure	Flat	Flat	Flat	Flat

Maier-Hein et al. [13] discussed the lack of success stories in the application of machine learning to surgery, and contrasted it to success in other medical fields, such as radiology and dermatology. This was directly attributed to the lack of quality annotated data, representative of the surgery domain, and the small size and limited representation of currently available datasets were reported to be major problems. One available labelled surgical tool dataset, while useful, provides images of only four tools [10]. Similarly, the currently available surgical tool datasets with a larger number of tools do not offer a sufficiently large range nor are they arranged hierarchically (Table 1). Kohli et al. [9] highlighted the lack of data for medical image evaluation with machine learning, and described current research as being "data starved" in this area. Current research focuses on convolutional neural networks (CNNs) trained on small medical datasets and the actual detection of less than fifty types of tools [2]; however, there are many thousands of surgical instrument types in circulation [20]. Clearly a new approach is required to handle this volume and variety of surgical tools. To help in addressing these challenges, we created a new surgical tool dataset named **HOSPITools**, short for "**H**ierarchically **O**rganised **S**urgical **P**rocedure **I**nstruments and **Tools**". This dataset offers a wide range of tools, and we evaluate its performance with different deep learning methods and techniques.

2 Class Hierarchies and Training Strategies

Image features learned by CNNs have been used extensively to classify images, or to retrieve images that are visually similar to a query image [4]. While deep CNNs are extremely effective in object classification and recognition, classification of fine-grained classes and discrimination between classes with relatively minor differences is a challenge [19]. This is a significant problem for our work, since many surgical tools are visually similar and often differ in minor, subtle and hard to discern ways. An approach that can potentially improve classification or retrieval performance for such fine grained classes is to embed prior knowledge of the classes or class hierarchies into the model [7]. Class hierarchies share knowledge of relationships in the ground truth class label arrangements, as opposed to class labels in a flattened arrangement where every class is assumed independent and unrelated, and incorporating this information into the model can potentially lead to better classification and retrieval performance.

The main challenge, as highlighted by Narayana et al. [15], lies in mapping images and labels to a shared latent space where embeddings that correspond to a similar semantic (not just visual) concepts lie closer to each other than embeddings corresponding to different semantic concepts. They addressed this problem by first constructing a semantic embedding space based on prior domain knowledge and then projecting image embeddings onto this fixed semantic embedding space. Their model ensured that distance between image embeddings were similar to corresponding class embedding distances in the semantic embedding space [15]. Barz and Denzler [4] computed class embeddings by a deterministic algorithm based on prior domain knowledge encoded in a hierarchy of classes – this was a novel feature level approach that mapped image embeddings to semantic embeddings, and successfully incorporated class information and semantic relationships into a deep learning model. The semantic embeddings of image features were shown to result in a model that was much more invariant against superficial visual differences such as colour and shape [4], and we therefore experiment with this method for our project.

The most common loss function used in the training of CNNs is the categorical cross-entropy loss in conjunction with a softmax activation, also known as the softmax loss [4, 23].

$$\mathcal{L}_{CCE} = -\sum_{i=1}^{k} c_i \log\left(\hat{c}_i\right) \tag{1}$$

In Equation 1, $\hat{c}_i$ represents the probability score for class c_i. This training strategy separates the classes, but it may not be sufficient for fine grained classification tasks [4]. The center-loss was therefore designed to increase the separation of classes while minimizing the distances between samples from the same class, and was defined as [23]:

$$\mathcal{L}_{center-loss} = \frac{1}{2}\sum_{i=1}^{k} \left\|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\right\|_2^2 \tag{2}$$

In Equation 2, $\boldsymbol{x}_i$ represents the center of the i^{th} class and $\boldsymbol{c}_{y_i}$ the deep feature vectors for each class. A multiple loss training strategy was used where the center-loss was employed to pull the deep features of the same class to their centers, while the softmax loss forced the deep features of different classes apart [23]. A combination of losses was also employed by Barz and Denzler [4], who used a classification loss along with an embedding loss designed to maximise the cosine similarity or the inner product between the image features and the embeddings of their classes. This correlation or cosine loss function was defined as:

$$\mathcal{L}_{CORR} = \frac{1}{k}\sum_{i=1}^{k} \Big(1 - \psi\left(I_i\right)^{\top} \varphi\left(c_{y_i}\right)\Big) \tag{3}$$

In Equation 3, φ defined the class embedding function, ψ the embedding function for image I, and $^{\top}$ was the dot product. Another important distance based loss is the mean squared error (MSE) loss, defined for class c_i as:

$$\mathcal{L}_{MSE} = \frac{1}{k}\sum_{i=1}^{k} \left(c_i - \hat{c}_i\right)^2 \tag{4}$$

We evaluate our dataset with these training strategies and loss functions.

3 Methodology

In this section, we describe the HOSPITools dataset, and we experiment with different strategies to train CNNs using this dataset. We believe that this dataset can be an important resource for AI and machine learning research on surgical tool management, and we use our experience with CNN training strategies to try to improve its structure and organisation.

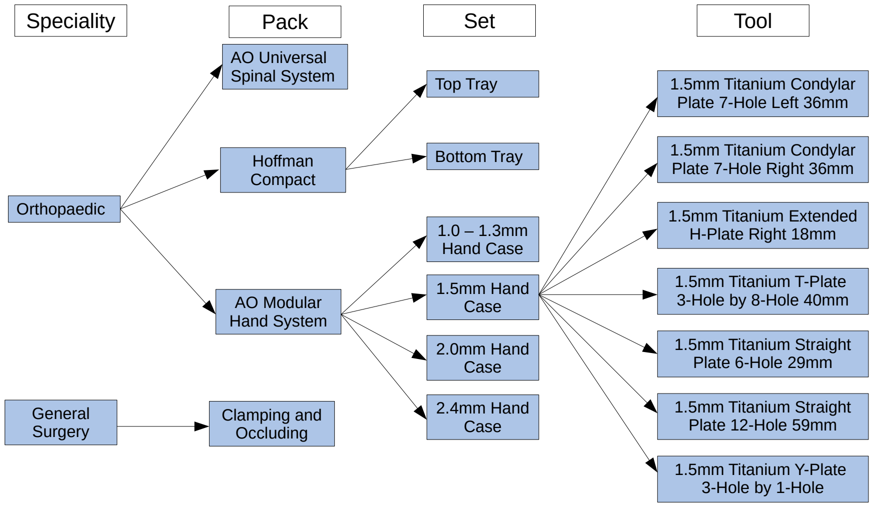
Speciality
Pack
Set
Tool
AO Universal Spinal System
Top Tray
1.5mm Titanium Condylar Plate 7-Hole Left 36mm
Hoffman Compact
Bottom Tray
1.5mm Titanium Condylar Plate 7-Hole Right 36mm
Orthopaedic
1.0 – 1.3mm Hand Case
1.5mm Titanium Extended H-Plate Right 18mm
1.5mm Hand Case
1.5mm Titanium T-Plate 3-Hole by 8-Hole 40mm
AO Modular Hand System
2.0mm Hand Case
1.5mm Titanium Straight Plate 6-Hole 29mm
General Surgery
Clamping and Occluding
2.4mm Hand Case
1.5mm Titanium Straight Plate 12-Hole 59mm
1.5mm Titanium Y-Plate 3-Hole by 1-Hole

Fig. 2. Surgical Tool Dataset Structure

3.1 Surgery Dataset

We developed our surgical dataset based on an hierarchical structure – speciality, pack, set and tool – as shown in Fig. 2. We captured RGB images of surgical tools using a DSLR camera, and manually arranged the images hierarchically in the dataset. We took these pictures on site in a major hospital, with the surgical tools currently in use. Image backgrounds were essentially flat colours, even though different backgrounds were used. Illumination sources included natural light – direct sunlight and shaded light – LED, halogen and fluorescent lighting. Distances of the camera to the object ranged from 60 to 150 cms. We focused on two specialities – Orthopaedics and General Surgery – for the initial stages of development of the dataset. The former speciality offers a wide range of instruments, implants and screws, while the latter covers the most common instruments used across all open surgery. We propose to add images of tools used in all 14 surgical specialities reported by the American College of Surgeons [1] in a phased manner as we develop this dataset. Our initial dataset consisted of 15,522 images across all hierarchies, with 11,712 images in the training set and 2,810 images in the validation set. We reserved a further 1,000 images for the test set, which the models did not see during training. While the average class size was 74 images, the range was from 139 images to as low as 10 images. This allowed us to evaluate the performance of the CNN training strategies with low class frequencies, and to explore how the dataset could be optimally structured with minimum images per class required for good performance.

3.2 Surgery Hierarchy

While it was relatively straightforward to train a baseline classifier using only the images and labels, some of our other strategies required additional information to be provided to the model. We therefore created a four level hierarchy in the surgery tool dataset, which consisted of speciality (2 classes), pack (12 classes), set (35 classes) and tool (360 classes) levels. The hierarchy was detailed in an indented tree format, which we then converted into "child-parent" tuples, as discussed by Barz and Denzler [4]. Dictionaries mapping class labels to lists of parent class labels and to child class labels in the hierarchy were created, and also a dictionary mapping hypernym identities of each element (class) to depths in the tree. We also developed lists of node identities, commencing with the direct hypernym of the given element and ending with the root node.

We only considered the taxonomic or hierarchical relationship between our classes in our work. The easiest relation is the "is-a" relation, which allows the specification of a hierarchical structure [4]. Hierarchies, most commonly represented as tree structures, provided us with an effective tool to organise and present the relationships and prior knowledge in our classes. In our tree structure, each class or node has just one parent class and distance was defined in terms of the length of the shortest path between two classes $\mathbf{c}_i, \mathbf{c}_j$. The dissimilarity of the classes $d_{\mathcal{G}}$ and the semantic similarity $s_{\mathcal{G}}$ was defined as [4]:

$$\begin{gathered} d_{\mathcal{G}} = \frac{\text{height}(\text{LCS}(c_i, c_j))}{\text{height }(\mathcal{G})} \\ s_{\mathcal{G}}(c_i, c_j) = 1 - d_{\mathcal{G}}(c_i, c_j) \end{gathered} \tag{5}$$

In Equation 5, LCS stands for lowest common subsumer – a class $\mathrm{c_i}$ was a subsumer of $\mathrm{c_j}$ if $\mathrm{c_j}$ was a descendant of $\mathrm{c_i}$ – and height $(\mathcal{G})$ is the height of the entire hierarchy. Using this, we obtained similarity measures in the range (0, 1), where "1" represented the maximum similarity (no distance) between classes. This information can then be used to train a CNN for image classification and retrieval [6], as will be shown in the next section.

3.3 CNN Training Strategies

We used the well researched and widely used ResNet-50 [8] for all our experiments. We computed the channel mean and standard deviation of the images in the training set, and used it to normalise the data. We resized the original 6000×4000 pixel images to 150×100 pixels, and used multiple data augmentation techniques, including flipping, scale augmentation and random cropping, to add diversity to the training data [8]. We evaluated the following experiments for our image classification and retrieval tasks, including hierarchy-based semantic image embeddings, based on prior work by Barz and Denzler [4]:

Baseline Classifier : As a baseline, we used a standard ResNet-50 and the features extracted from the layer before the final classification layer of the network architecture. We used categorical cross entropy as the loss function.

Center-loss : We used the ResNet-50 architecture and trained it with both center-loss and softmax loss, following Wen et al. [23]. We maintained the center-loss weight at 0.1 – this value was used to balance the two loss functions. Wen et al. [23] experimented with changes of this weight from 0 to 0.1; with the weight at 0, or only using softmax loss, they obtained a poor result but performance was relatively unchanged across other variations of this weight.

MDS Embeddings : We computed embeddings in 360 dimensional space so that the distances of class embeddings corresponded to their semantic dissimilarity (Eq. 5) using classical multidimensional scaling (MDS). We used the MSE loss in this distance based approach.

Sphere Embeddings : We calculated a "360-by-360" matrix specifying the distance between each pair of classes, based on the dissimilarity score of the two classes (Eq. 5). Following Barz and Denzler [4], with the first class at the origin, the second class was located at an offset along the first axis by the specified distance. We then placed all remaining classes in an iterative manner at an intersection of the hyperspheres centered at existing classes, with the radii set at the distance of the new class. We used the MSE loss in this training strategy.

Unitsphere Embeddings : The problem statement is: Given a distance matrix D, we wanted to place the set of points on a unit hypersphere which produce the same distance matrix. We used Eq. 5 to calculate similarities and the following equation to place class embeddings, where φ defined the class embedding function and $^\top$ was the dot product [4]:

$$\begin{aligned} &\varphi\left(c_i\right)^\top \varphi\left(c_j\right) = s_{\mathcal{G}}\left(c_i, c_j\right) \\ &\left\|\varphi\left(c_i\right)\right\| = 1 \end{aligned} \tag{6}$$

Equation 6 stated that the correlation of class embeddings should equal their similarity. The second function ensured that the L2-norm embeddings were on the unit hypersphere, and the dot product was then used as a substitute for the Euclidean distance [5]. The network was trained to minimise the difference between image representations and the embeddings of their respective class as per the guidelines of Barz and Denzler [4] using a combined loss $\mathrm{L}_{\mathrm{CORR+CLS}} = \mathrm{L}_{\mathrm{CORR}} + \lambda\ \mathrm{L}_{\mathrm{CLS}}$. Since we desired that the embedding loss $\mathrm{L}_{\mathrm{CORR}}$ dominated learning, we set λ to a very low value (0.1) in our experiments (a similar value was used in the center-loss strategy).

We tried two different learning schedules for our training, a standard ResNet training schedule and Stochastic Gradient Descent with Cosine Annealing and Warm Restart (SGDR) [12]. While we tested these learning rates on each of our strategies, we only present the SGDR results since they are much better. This schedule implemented warm restarts, where in each restart the learning rate was initialized to a new value, scheduled to decrease over the cycle. The initial learning rate at the beginning of each cycle was 0.1, decreasing to a minimum of 10^{-6} using cosine annealing based on number of epochs since the last restart [12]. The first cycle was set at 12 epochs, the multiplier for cycle length was set at 2, and training was for 5 cycles or 372 epochs [4].

Table 2. Classification Results

SGDR	Accuracy	Top-5 Accuracy	Hierarchical Accuracy	F1-Score
Classifier	0.84	0.98	0.80	0.81
Center-loss	**0.88**	**1.00**	0.83	**0.86**
MDS	0.83	0.96	0.79	0.80
Spheres	0.85	0.98	0.81	0.82
Unitsphere	**0.88**	0.99	**0.84**	**0.86**

Table 3. Retrieval Results (WUP)

SGDR	HP@1	HP@10	HP@50	HP@100	AHP	mAP
Classifier	0.84	0.68	0.56	0.54	0.83	0.47
Center-loss	0.91	0.81	0.65	0.60	0.84	0.76
MDS	0.90	0.87	0.87	0.87	0.95	0.73
Spheres	0.90	0.88	0.89	0.89	0.97	0.76
Unitsphere	**0.93**	**0.91**	**0.91**	**0.91**	**0.98**	**0.84**

3.4 Metrics Reported

We report the Accuracy, Top-5 Accuracy, Hierarchical Accuracy and F1-Score for the classification performance. For the retrieval tasks, we report the hierarchical precision of the nearest neighbour search performed on different image embeddings – HP@k for different k values, Average Hierarchical Precision (AHP) and Mean Average Precision (mAP). The Hierarchical Precision at k (HP@k) is a generalization of Precision@k which takes class similarities into account [7], and we report this for k at 250. This is calculated by the sum of similarities between query image class and retrieved image class over the top k retrieval results, divided by the maximum possible sum of top-k class similarities. Average Hierarchical Precision is defined by the area under the hierarchical precision curve, with the optimum normalized at 1.0. The Mean Average Precision, which does not consider class similarities, is also reported for comparison.

Class similarity is reported by the Wu-Palmer similarity metric ("WUP"), which considered the height and position of classes relative to each other in the tree – classes further from the root with a common parent tend to be more semantically similar. The WUP measure was calculated from equation 5.

4 Experiments and Results

Classification performance is good across the board, and there is no significant improvement in basic accuracy by including hierarchical information, as shown in Table 2. However, the biggest impact of including prior information and in the embedding strategies is found in the retrieval task, as shown in Table 3. Retrieval of single images is good for all models tested, but as the number of similar images

Table 4. Class Frequency Classification Results

Images per Class	Class	F1-Score
139	Curved Mayo Scissors	0.96
117	7-inch Metzenbaum Scissors	0.94
15	0.76mm Drill Bit with 10mm Stop Mini QC	0.17
18	0.76mm Drill Bit with 12mm Stop Mini QC	0.22
13	Universal Spinal System Holding Sleeve	1
11	Jacobs Chuck	1

retrieved increases, there is a definite advantage in terms of the embedding strategies. There is a significant drop in accuracy with increase in the k value with the Classifier and Center-loss models, but embedding with the MDS, Spheres and Unitsphere strategies demonstrates a consistent performance across different k values. Since the number of images per class is low, smaller k values retrieve images from exactly the same category as the query but as k increases, images are retrieved from outside the direct class. This is where the incorporation of semantic information excels, retrieving images from semantically similar classes even at higher k values. Semantic information significantly improves the quality of content-based image retrieval, by retrieving images that are both visually and semantically similar. Incorporating prior knowledge about class similarities by mapping class embeddings in semantic space appears to facilitate better learning by the CNN, thereby leading to better retrieval results. Organising the surgical tool dataset in the form of a hierarchical structure, and providing additional information about the taxonomic or hierarchical relationship between our classes, is therefore conclusively demonstrated to be an approach that leads to better performance, at least for the image retrieval tasks.

4.1 Does Size Matter?

The original images were captured at 6000 by 4000 pixels, on the assumption that finer detail could be captured and it would be easier to down-sample the images than to up-sample. Down-sampling was done to improve data handling, storage and processing, and we evaluated the effects of resizing images in the pre-processing pipeline on the CNN performance. We experimented with images of 600 by 400 pixels, with 300 by 200 pixels, with resizing the images to 224 by 224 with padding, and with image size of 150 by 100 pixels. There was no degradation in performance even at the smaller sizes, and so we implemented our training at an image size of 150 by 100 pixels, with random cropping of 100 by 100 pixels during augmentation. Our findings can be contrasted with the work of Sabottke and Spiele [17], who examined image resolution variations on CNN performance for radio-graphic images. While they did find some performance differences, this was relevant only when finer details needed to be captured for the diagnosis-specific tasks. For our objects of interest, image size variances do not appear to be as significant but this is a promising avenue for future work.

4.2 Class Frequencies

The class frequencies for the training set were averaged at 74 images, with a range from 10 to 139 images per class. While additional images were available, we wanted to test performance with different class frequencies. This was difficult to analyse – we obtained good classification results (Unitsphere strategy) even with 11 images per class, while much higher class frequencies did not yield the best results (Table 4). Clearly the number of images required for good performance depends on the particular tool and its distinctiveness in the dataset. An initial benchmark – at least for this dataset, for classification tasks, with the prior hierarchy information, and for these types of tools – does appear to be at least 40 images per class but this is not conclusive. As more cluttered images in realistic and messy settings are added, more images will be required to maintain accuracy and predictive performance. We will revisit this as we expand the scope and scale of our dataset.

5 Conclusions and Future Work

We developed a new surgical tool dataset – **HOSPITools** – and used it to test different CNN learning strategies. We demonstrated that the hierarchical nature of surgical tool classes could be used to make improved predictions. We also used the training to explore how the dataset should be structured and to evaluate some design parameters. This was a proof of concept for accurate recognition of surgical tools by utilising the hierarchical nature of the classes, and this solution can be used for intelligent management of surgical tools in a hospital.

We will continue to improve the dataset, with a view to making it publicly available for AI and machine learning research. We will address threats to the validity and utility of our work by adding images from more of the 14 surgical specialities, and by including greater coverage and variety in each speciality. We will include images with greater occlusions, reflections, illumination changes, the presence of blood, tissue and smoke, varied backgrounds, and from different modalities such as video, infrared and depth images. Open surgery and laparoscopic surgery images need to be sourced if possible, including live surgeries. If we can do this, then the surgery tool dataset can potentially be a valuable resource for the AI and machine learning communities.

References

1. ACS. What are the surgical specialties? https://www.facs.org/education/resources/ medical-students/faq/specialties, 2021. Accessed: 15/2/2021.
2. H. Al Hajj, M. Lamard, P. H. Conze, et al. Cataracts: Challenge on automatic tool annotation for cataract surgery. *Medical Image Analysis*, 52: 24–41, 2019. https://doi.org/10.1016/j.media.2018.11.008.

3. M. Allan, A. Shvets, and T. et al. Kurmann. 2017 robotic instrument segmentation challenge. *ArXiv, abs/*, 1902:06426, 2019.
4. Bjorn Barz and Joachim Denzler. Hierarchy-based image embeddings for semantic image retrieval. In *In IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019. https://doi.org/10.1109/WACV.2019.00073.
5. Bjorn Barz and Joachim Denzler. Deep learning on small datasets without pre-training using cosine loss. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020. https://doi.org/10.1109/WACV45572.2020.9093286.
6. Clemens-Alexander Brust and Joachim Denzler. Not just a matter of semantics: the relationship between visual similarity and semantic similarity. *Lecture Notes in Computer Science, vol 11824*, 2019. https://doi.org/10.1007/978-3-030-33676-9_29.
7. J. Deng, A. Berg, and L. Fei-Fei. Hierarchical semantic indexing for large scale image retrieval. In *CVPR 2011, 785-792*, 2011. https://doi.org/10.1109/CVPR.2011.5995516.
8. K. He, X. Zhang, S. Ren, et al. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition. Washington (DC)*. IEEE Computer Society, 2016. https://doi.org/10.1109/CVPR.2016.90.
9. Marc D. Kohli, Ronald M. Summers, and J. Raymond Geis. Medical image data and datasets in the era of machine learning – white paper from the 2016 C-MIMI Meeting Dataset Session. *Journal of Digital Imaging (2017) 30.*, page 392399, 2017. https://doi.org/10.1007/s10278-017-9976-3.
10. Diana Martins Lavado. Sorting surgical tools from a cluttered tray - object detection and occlusion reasoning. Master's thesis, University of Coimbra, Portugal, 2018.
11. T. Leppanen, H. Vrzakova, R. Bednarik, et al. Augmenting microsurgical training: Microsurgical instrument detection using convolutional neural networks. In *IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS). Karlstad*. Sweden, 2018. https://doi.org/10.1109/CBMS.2018.00044.
12. Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations*, 2017.
13. L. Maier-Hein, M. Eisenmann, D. Sarikaya, K. Marz, et al. Surgical data science - from concepts to clinical translation. *ArXiv, abs/2011.02284*, 2020.
14. J. M. Mhlaba, E. W. Stockert, M. Coronel, and A. J. Langerman. Surgical instrumentation: The true cost of instrument trays and a potential strategy for optimization. *Journal of Hospital Administration*, 4:6, 2015. https://doi.org/10.5430/jha.v4n6p82.
15. P. Narayana, A. Pednekar, A. Krishnamoorthy, K. Sone, and S. Basu. HUSE: Hierarchical universal semantic embeddings. *ArXiv, abs/1911.05978.*, 2019.
16. T. Ross, A. Reinke, and P. M. et al. Full. Robust medical instrument segmentation challenge. *ArXiv preprint*, 2019.

17. C. F. Sabottke and B. M. Spiele. The effect of image resolution on deep learning in radiography. *Radiology: Artificial Intelligence*, 2(1), 2020. https://doi.org/10.1148/ryai.2019190015.
18. D. Sarikaya, J. J. Corso, and K. A. Guru. Detection and localization of robotic tools in robot-assisted surgery videos using deep neural networks for region proposal and detection. *IEEE Transactions on Medical Imaging*, 36 (7):1542–1549, 2017. https://doi.org/10.1109/TMI.2017.2665671.
19. F. Setti. To know and to learn about the integration of knowledge representation and deep learning for fine-grained visual categorization. In *13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, 2018. https://doi.org/10.5220/0006651803870392.
20. Sklar. Surgical instruments: The introductory guide. Sklar Instrument, West Chester, PA., 2016.
21. E. W. Stockert and A. J. Langerman. Assessing the magnitude and costs of intraoperative inefficiencies attributable to surgical instrument trays. *Journal of the American College of Surgeons*, 219(4):646–655, Oct 2014. https://doi.org/10.1016/j.jamcollsurg.2014.06.019.
22. A. P. Twinanda, S. Shehata, D. Mutter, et al. Endonet: A deep architecture for recognition tasks on laparoscopic videos. *IEEE Transactions on Medical Imaging*, 36:86 – 97, 2017. https://doi.org/10.1109/TMI.2016.2593957.
23. Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *In European Conference on Computer Vision (ECCV)*, 2016. https://doi.org/10.1007/978-3-319-46478-7_31.
24. Z. Zhao, S. Voros, Y. Weng, F. Chang, and R. Li. Tracking-by-detection of surgical instruments in minimally invasive surgery via the convolutional neural network deep learning-based method. *Computer Assisted Surgery*, 22:26–35, 2017. https://doi.org/10.1080/24699322.2017.1378777.