tvst

Article

Analysis of Cataract Surgery Instrument Identification Performance of Convolutional and Recurrent Neural Network Ensembles Leveraging BigCat

Nicholas Matton[1], Adel Qalieh[2], Yibing Zhang[2], Anvesh Annadanam[2], Alexa Thibodeau[2], Tingyang Li[3], Anand Shankar[3], Stephen Armenti[2], Shahzad I. Mian[2], Bradford Tannen[2], and Nambi Nallasamy[2,3]

[1] Department of Computer Science, University of Michigan, Ann Arbor, MI, USA
[2] Kellogg Eye Center, Department of Ophthalmology and Visual Sciences, University of Michigan, Ann Arbor, MI, USA
[3] Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI, USA

Correspondence: Nambi Nallasamy, Kellogg Eye Center, University of Michigan, 1000 Wall St, Ann Arbor, MI 48105, USA. e-mail: nnallasa@med.umich.edu

Received: November 26, 2021
Accepted: March 2, 2022
Published: April 1, 2022

Keywords: machine learning; cataract surgery; computer vision; surgery tool presence

Citation: Matton N, Qalieh A, Zhang Y, Annadanam A, Thibodeau A, Li T, Shankar A, Armenti S, Mian SI, Tannen B, Nallasamy N. Analysis of cataract surgery instrument identification performance of convolutional and recurrent neural network ensembles leveraging BigCat. Transl Vis Sci Technol. 2022;11(4):1, https://doi.org/10.1167/tvst.11.4.1

Purpose: To develop a method for accurate automated real-time identification of instruments in cataract surgery videos.

Methods: Cataract surgery videos were collected at University of Michigan's Kellogg Eye Center between 2020 and 2021. Videos were annotated for the presence of instruments to aid in the development, validation, and testing of machine learning (ML) models for multiclass, multilabel instrument identification.

Results: A new cataract surgery database, BigCat, was assembled, containing 190 videos with over 3.9 million annotated frames, the largest reported cataract surgery annotation database to date. Using a dense convolutional neural network (CNN) and a recursive averaging method, we were able to achieve a test F1 score of 0.9528 and test area under the receiver operator characteristic curve of 0.9985 for surgical instrument identification. These prove to be state-of-the-art results compared to previous works, while also only using a fraction of the model parameters of the previous architectures.

Conclusions: Accurate automated surgical instrument identification is possible with lightweight CNNs and large datasets. Increasingly complex model architecture is not necessary to retain a well-performing model. Recurrent neural network architectures add additional complexity to a model and are unnecessary to attain state-of-the-art performance.

Translational Relevance: Instrument identification in the operative field can be used for further applications such as evaluating surgical trainee skill level and developing early warning detection systems for use during surgery.

Introduction

Cataract surgery is one of the most commonly performed surgical procedures in the world and is a fundamental part of ophthalmology training. Complication rates for cataract surgery are low and have decreased with improved phacoemulsification technology and training methodology.[1] These improvements have primarily come in the form of improved anterior chamber stability and surgical simulators, respectively. Providing consistent, objective feedback on surgical quality remains a challenge, however. Verbal intraoperative feedback can be difficult given the use of topical anesthesia and limited patient sedation, and providing feedback after surgery can be difficult given the premium placed on time in the operating room and the need to move between surgical cases efficiently. There are also limited options available for validated tools available for cataract surgery evaluation. Moreover, although expert surgeons are able to provide qualitative feedback, quantitative feedback may be of

translational vision science & technology

Table 1. Summary of Cataract Surgery Video Databases With Number of Videos and Number of Annotated Frames Where Reported

Citation	Number of Videos	Content	Annotations
Quellec et al., 2014[15]	186	Surgical sequence	Surgical phase
Al Hajj et al., 2017[2]	30	Surgical sequence	Instrument appearance/disappearance
Schoeffmann et al., 2018[16]	101	Surgical sequence	Surgical phase
Yu et al., 2019[17]	100	Surgical sequence	All frames
Zang et al., 2019[18]	52	Surgical sequence	Select frames (5,010)
Morita et al., 2020[19]	302	Surgical sequence	Select frames (12,634)
Al Hajj et al., 2019[3]	50	Surgical sequence	Instrument contact with eye
Matton et al.	190	Surgical sequence	All frames (3,946,653)

value in improving surgical performance, particularly with regard to steps such as the capsulorrhexis and nucleus disassembly. In addition, there are limitations in comparing trainee performance over the length of the training program and across trainees.

To move toward the goal of providing objective feedback on surgical performance, the automated identification of instruments within the surgical field is an essential step. The ordering, duration, and location of surgical instruments at different points throughout a surgery may indicate how well a surgeon performed or whether there were complications during the surgery. By creating a machine learning model that can accurately identify when a surgery tool is being used during a surgery, we take an important step toward creating an accurate surgical assessment tool. Although detecting surgical instruments has been attempted before, no previous attempts have had the ability to train on the large amount of annotated cataract surgery data we have gathered (Table 1). Where previous studies have reported only area under the receiver operator characteristic curve (AUROC) values as their primary performance metric,[2,3] we report accuracy, precision, recall, and F1 scores. These metrics are expected to be more indicative of model performance, particularly in the setting of class imbalance.[4,5] Additionally, we report the number of frames used in our training, testing, and validation datasets, giving a more specific quantification of the amount of data used. We also report the number of parameters and the inference times of our top performing models, making clearer the tradeoff between model complexity and speed.

Recent approaches in the CATARACTS challenge use a combination of convolutional neural networks (CNNs) and post-prediction smoothing techniques to identify instrument presence in videos of cataract surgery.[3] These methods combine, in some cases, up to four different CNNs followed by post-processing smoothing techniques in order to attain state-of-the-art performance. Although such methods achieve top-tier results, the architectures used were exceptionally large, and investigations did not consider time, space, and expense tradeoffs. A method reported by al Hajj et al.[3] involved a novel CNN that processes sequences of images instead of processing each image individually, analogous to smoothing techniques that process multiple images at a time.[2] This approach, however, did not achieve the top-tier results seen in more recent works.[3] It is unclear whether this was due to limitations of the architecture itself or the training dataset used.

To create a real-time instrument detection model for incorporation into a surgical assessment system, limitations on architecture complexity and size must be considered. In this article, we show that to attain top performance, it is not necessary to create a complex system of neural networks. By training on a large, annotated dataset and using a single CNN architecture, we create a model that is a fraction of the size of many previous architectures while achieving state-of-the-art results.

Methods

Data Collection

Video recordings of cataract surgeries performed by attending surgeons at University of Michigan's Kellogg Eye Center were collected between 2020 and 2021. Institutional review board approval was obtained for the study (HUM00160950), and it was determined that informed consent was not required because of its retrospective nature and the anonymized data used in this study. The study was carried out in accordance with the tenets of the Declaration of Helsinki. All surgical videos were recorded using Zeiss high definition one-chip imaging sensors integrated into ceiling-mounted Zeiss Lumera 700 operating microscopes (Carl Zeiss Meditec AG, Jena, Germany). The imaging sensor received light split from the optical pathway of the primary surgeon's scope head and the signal was recorded to a Karl Storz AIDA recording device in full high-definition (1920 × 1080) resolution (Karl Storz SE & Co. KG, Tuttlingen, Germany). All surgeries were performed using the Alcon Centurion phacoemulsification machine (Alcon AG, Fort Worth, TX, USA). Femtosecond laser cataract surgeries and complex cataract surgeries (those qualifying for Current Procedural Terminology code 66982) were excluded. Cases with incomplete recordings were also excluded. Segments from before surgery and after surgery were trimmed, but video during surgery was otherwise completely unedited. The source resolution was 1920 × 1080 pixels at a frame rate of 30 frames/sec. Frame by frame instrument annotations were performed by a contracted third-party annotation services provider (Alegion Inc., Austin, TX, USA). Alegion's proprietary workflow was followed, which included (1) training of Alegion labeling technicians by NN, (2) two rounds of instrument labeling validation by NN on videos not included in the final dataset, and (3) final automated checks on received annotations to ensure that each video frame had corresponding instrument annotations. Alegion's proprietary video labeling platform was used by their labeling technicians to perform the annotations, and annotations were provided in JavaScript object notation (JSON) format. We have written software that converts the Alegion-structured JSON–formatted annotations to the open and well-documented COCO format, which is widely supported by open-source image labeling software such as Computer Vision Annotation Tool (CVAT) and labelImg. This enables one to evaluate and build on existing annotations with open-source labeling platforms, ensuring quality and reproducibility for future research. Through use of this third-party annotation services provider, it was ensured that no surgeons involved in the study were involved in the manual annotation of videos included in the dataset. A total of 208 videos were selected for annotation of instrument presence ground truth for every frame. One hundred ninety videos passed annotation validation checks to ensure appropriate and complete annotations for all available frames. The resulting dataset of 190 surgical videos and their annotations was termed BigCat. Over the set of surgical videos, 10 distinct instruments (listed in Supplementary Table S1) were annotated for their presence or absence with a binary designation for each instrument for each frame. Table 1 provides a comparison of BigCat with other reported cataract surgery video datasets.

Data Preprocessing

The hydrodissection cannula and the 27-gauge cannula labels were combined into a general "cannula" label, as these instruments at our institution were visually similar. Video frames were resized to 480 × 270 pixels to improve the speed of the training and inference processes. In order to augment the data for training, transformations were randomly applied to input images.[6] This was intended to improve the generalizability of the models studied. The types of transformations applied were rotations, shifts, shears, zooms, horizontal flips, and rescales. Of the 190 videos that passed validation checks 114 videos (2,282,382 frames) were allocated for training, 38 videos (838,005 frames) were allocated for validation, and 38 videos (826,266 frames) were held out for testing.

Model Development

We sought to evaluate the instrument identification performance of CNNs individually or in an ensemble, with or without a postprocessing technique (Fig. 1). This approach was designed to quantify the tradeoffs between model complexity and performance. The problem itself was posed as a multilabel classification problem with the nine aforementioned classes. To speed up model development, we did not use our full dataset when training and validating these models. Instead, we sampled 100 random batches of size 32 without replacement for each epoch, and we subsequently trained for 200 epochs. This amounted to exposure to approximately 28% of our training data and took approximately six hours to run.

The first algorithm considered consisted of a CNN, a dense neural network (NN), and a sigmoid function. The CNN was used to draw spatial patterns from the input images, whereas the dense NNs were meant to make predictions on the input images. The sigmoid mapped these predictions into a probability between 0 and 1. The output was a set of probabilities that represent the confidence that each surgery tool was present within a given input frame. Our final predictions were gathered by thresholding the output probabilities such that any output probability over a certain value for a surgery tool will result in the model predicting that tool is in frame. The specific value is a hyperparameter that we tune to optimize performance. The CNNs used were Densenet169 and Inception-ResNet-V2.[7,8] Densenet169 was used because of its densely connected network used to mitigate the vanishing gradient problem and promote feature reuse. Inception-ResNet-V2 was used because of its state-of-the-art performance on the ImageNet dataset.

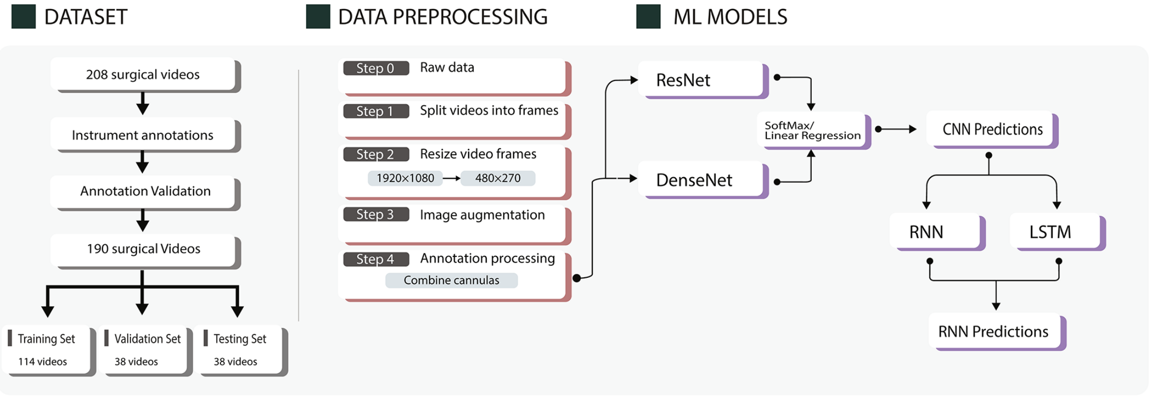

Figure 1. Summary of the model architecture created for surgery tool detection. If RNNs were not used, the model made predictions after the CNNs were combined using Softmax or Linear Regression. If only a single CNN was used (DenseNet or ResNet), we returned the CNN predictions unchanged.

To attempt to improve on the results from using each CNN individually, the second approach used was a strategy called ensemble classification. The ensemble classifier took in the input video frames and passed them to the two CNN models, which each made their own predictions for the likelihood that a tool was in frame. This strategy is meant to draw on the strengths of both networks when making predictions. We combined the probabilities of the two CNNs in two different ways. Our first approach was a simple arithmetic mean of the probabilities. Our second approach was to use a linear regression model to combine the probabilities.

To incorporate time dependencies and address smoothness of the CNN predictions, the third approach involved the addition of recurrent neural networks (RNNs) on top of our CNN models. Two different RNNs were layered on top of the CNN models. The first of the RNNs implemented was a long short-term memory (LSTM) network followed by a dense NN layer and a sigmoid function. The second was a fully connected SimpleRNN followed by a dense NN layer and a sigmoid function. These RNNs were layered on top of the CNN ensemble, as well as over the CNNs individually.

We also implemented a recursive averaging algorithm to smooth the predictions from our CNNs. We average the predictions for each tool across a five-frame window. This window consists of the current frame, two frames prior to and two frames after the current frame. For the frames prior, we use the averaged prediction values that were created in the last two iterations of averaging. We take our final output probability as this average across the five frames. This approach uses negligible processing power with respect to our models.

Model Evaluation

Model performance was evaluated using a wide range of metrics. These included class accuracy, recall, precision, F1 score, and AUROC. For this problem in which each tool is used for only a small fraction of the entire surgery, accuracy and AUROC values may be inflated.[9,10] The F1 score, which captures both precision and recall values, offers a broader view of model performance.[11,12] We ran a grid search over learning rate and batch size on our model with the best validation metrics to optimize its performance. We tested learning rates of 1e-4, 1e-5, and 1e-6 and batch sizes of 16 and 32. We also ran an exhaustive search over the prediction confidence threshold value. Once finding the optimal hyperparameters, we trained our model over our entire dataset.

Statistical Analysis

Differences in model performance on the validation set were assessed using the Friedman test, followed by post hoc paired Wilcoxon signed-rank tests with Bonferroni correction.

Implementation

Data pipelines and machine learning models were developed and tested in Python 3.7.7 with TensorFlow 2.2.0 and Keras 2.3.0. All statistical analysis was performed using Python 3.7.7. Testing, including inference time measurements, were performed using a machine with 4 Nvidia RTX 2080 Ti GPUs. For each test run, we use two GPUs to load the model, to load the testing data, and to make inferences on the testing data.

Results

Dataset Characteristics

A final dataset consisting of annotated video recordings of 190 cataract surgeries performed by nine attending surgeons at University of Michigan's Kellogg Eye Center was gathered. The source resolution was 1920 × 1080 pixels at a frame rate of 30 frames/sec with an average duration of 692 seconds and standard deviation of 161 seconds (Fig. 2a). The average time with the paracentesis blade visible was nine seconds (1% of overall procedure video length) and was consistently at the beginning of the procedure (Fig. 2b). In contrast, the phacoemulsification handpiece was visible on average 241 seconds (35% of video length) and the irrigation/aspiration handpiece was visible on average 137 seconds (20% of video length) (Fig. 2c).

Model Performance

The validation performance for each model is presented in Supplementary Table S2. The Inception-ResNet-V2 and DenseNet169 models performed at the highest level while remaining low cost with respect to other architectures (Fig. 3). The Inception-ResNet-V2 achieved a validation F1 score of 0.9189 and validation AUROC of 0.9860 and contained 90,121,961 parameters. The DenseNet169 achieved a validation F1 score of 0.9273 and validation AUROC of 0.9905 and contained 63,763,529 parameters. A Friedman test for differences in the F1 scores among the models studied yielded a test statistic of 207.36 and a *P* value of 4.68e-39, indicating a difference among the models. Post hoc paired Wilcoxon signed-rank tests with Bonferroni correction demonstrated that the DenseNet169 model had no statistical difference in performance in comparison to the Inception-ResNet-V2 model, but performed statistically better than the two CNN-only ensembles (see Supplementary Table S3 for *P* values). These CNN ensembles were outperformed by DenseNet169 despite using more than 2.4 times the number of model parameters (153,885,490 vs. 63,763,529). DenseNet169 with recursive averaging performed statistically significantly better than all other models studied, including the models using RNNs (see Supplementary Table S4 for *P* values).

Our top performing model, DenseNet169 with recursive averaging, achieved a validation F1 score of 0.9322 and a validation AUROC of 0.9913. The additional resources needed for recursive averaging are nearly negligible with respect to the amount of processing time and memory usage.

We then ran a grid search across batch size and learning rate to optimize the performance of the DenseNet169 model. We found that a batch size of 32 and a learning rate of 1e-6 optimized performance for the DenseNet169 model. We also conducted an exhaustive search across our prediction threshold to optimize F1 score, and we found a threshold of 0.41 to optimize our performance. With these optimal hyperparameters, we then trained the DenseNet169 model on the full dataset for 6 epochs (approximately 30 hours of training time per epoch) and analyzed this final model on our testing data. This allowed us to achieve a testing F1 score of 0.9528 and a testing AUROC of 0.9985. Performance of our final model on the testing set is summarized in Table 2.

We also analyzed our model qualitatively. As can be seen in Supplementary Figure S1, which depicts the final model's instrument time course predictions and ground truth for a representative case, the predictions from our model appear similar to the actual instrument presence.

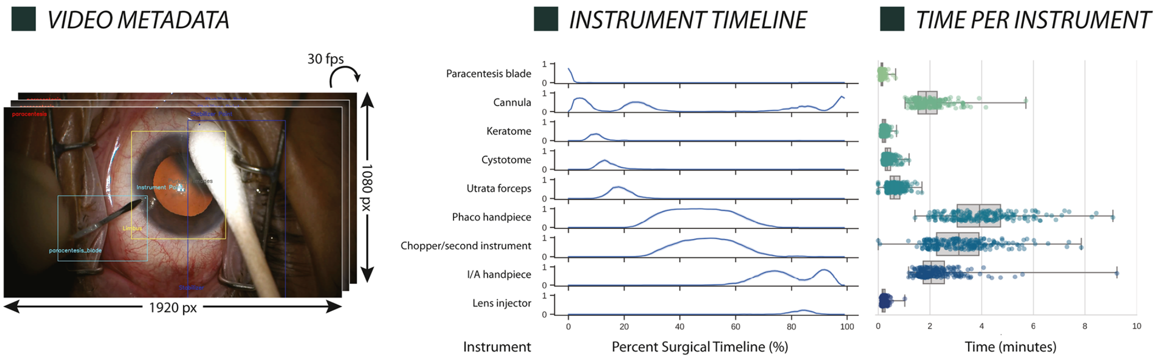

Figure 2. (a) Visual summary of the video metadata. (b) Density plot of each surgical instrument's use as a function of the percentage of the surgical video timeline averaged across 190 annotated videos. (c) Time per instrument summary statistics across 190 videos. Each point represents an individual video and the time given instrument was used in the recorded procedure.

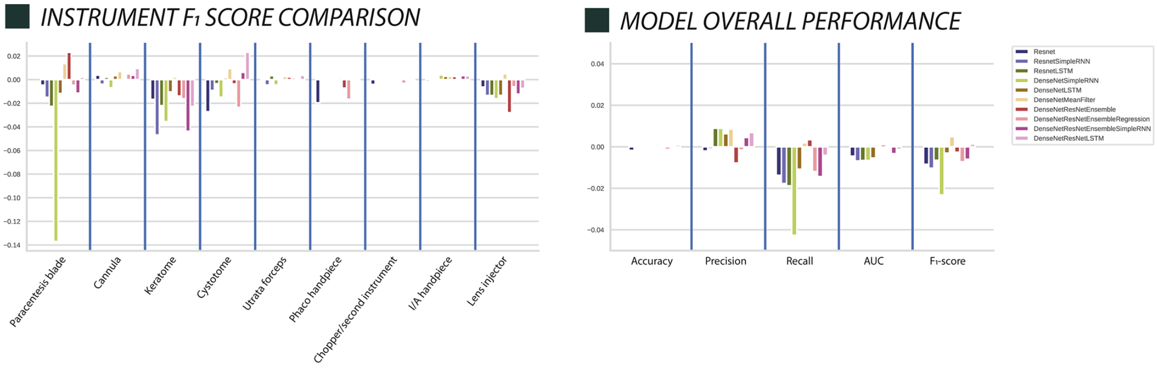

Figure 3. Summary of model performance relative to DenseNet. (a) Absolute difference in F1 score for instrument detection for each model relative to DenseNet. (b) Absolute difference in performance of each model relative to DenseNet performance for each metric.

Table 2. Class-Wise Test Metrics for Our Final Model (DenseNet169 Model With Recursive Averaging)

	Cystotome	Chopper	I/A Handpiece	Keratome	Lens Injector	Para Blade	Phaco Handpiece	Utrata Forceps	Cannula	Overall
Accuracy	0.9939	0.9891	0.9968	0.9987	0.9981	0.9980	0.9965	0.9949	0.9758	0.9935
Precision	0.9735	0.9906	0.9989	0.9582	0.9835	0.8794	0.9979	0.9903	0.9677	0.9711
Recall	0.8216	0.9707	0.9845	0.9721	0.9207	0.9653	0.9921	0.9210	0.8869	0.9372
AUROC	0.9985	0.9976	0.9999	0.9996	0.9993	0.9995	0.9999	0.9996	0.9929	0.9985
F1 Score	0.8911	0.9805	0.9917	0.9651	0.9511	0.9203	0.9950	0.9544	0.9256	0.9528

The full names of each instrument are defined in Supplementary Table S1.

Inference Time

We compared the average inference times of our top performing models. We used two GPUs to load each model and the testing data. The DenseNet169 model was fastest with an inference time of about 0.00598 seconds per frame (∼167 frames/sec). The ResNet was slightly slower with an inference time of about 0.00721 seconds per frame (∼143 frames/sec). The Ensemble classifier consisting of DenseNet169 and Inception-ResNet-V2 was slower with an inference time of about 0.0128 sec/frame (∼77 frames/sec).

Discussion

Intuitively, the tools that are in use during a surgery are important in the outcome of the surgery. It then follows that the ability to identify which tools are currently in use is an important first step in building a video-based surgery assessment tool. Our intention with this study was to develop an efficient model with state-of-the-art performance in instrument identification to enable downstream processing for more complex recognition and assessment tasks. Our 190 video dataset, BigCat, was gathered with a full 30 frame/sec frame rate and full 1920 × 1080 frame resolution. This equates to 3,946,653 full resolution video frames. Compared to other recently gathered cataract video datasets, the dataset we present here, BigCat, is orders of magnitude larger.[3] Many recent approaches downsample the frame rate of the videos considerably when training and testing on the data.[2,3] With BigCat, every frame is annotated with instrument presence data, allowing for use of the full 30 frames per second when training and testing our models. See Table 1 for a comparison of BigCat between other reported datasets.

We found the DenseNet169 architecture with recursive averaging to be the best performing model among those tested. Although DenseNet169, Inception-ResNet-V2, and the CNN ensemble architectures all achieve similar performances, the DenseNet169 performed slightly better with 30% fewer parameters than Inception-ResNet-V2 and 59% fewer parameters than the CNN ensemble, making it a more efficient choice. When trained on the full BigCat dataset, our final DenseNet169 model with recursive averaging achieved an overall test F1 score of 0.9528 and an overall test AUROC of 0.9985. Compared to the DResSys and Multi Image Fusion models, which achieved AUROC of 0.9971 and 0.977 respectively, these are state-of-the-art results.[2,3] DResSys performs at a similar level as our model with respect to AUROC; however, it uses a combination of an Inception-V4, a ResNet-50, and two NasNet-A models, making this architecture around four times larger than our DenseNet169 model with recursive averaging.[13,14] As mentioned above, the F1 scores for the DResSys and Multi Image Fusion models were not reported, precluding comparison of this metric. The F1 score is of particular importance for classification problems with significant class imbalance, such as instrument identification. In cataract surgery, each surgical instrument is used for only a small fraction of the surgery, causing a large disparity in the number of negative and positive examples for each instrument. The scarcity of positive examples may cause accuracy and AUROC values to be inflated. For example, a model that predicts the paracentesis blade is never in frame may still achieve a high accuracy and AUROC score because it will be correct for most frames. Additionally, previous studies have found the AUROC to be unreliable when discriminating among multiple high-performing models. Because the F1 score is not affected by the number of true-negative predictions, it is better able to avoid inflation caused by the imbalance between positive and negative examples inherent in the problem of cataract surgery instrument identification.[11,12]

The DenseNet169 model achieved an average inference time of 0.00598 sec/frame on the standard hardware described, which is equivalent to approximately 167 frames/sec. It is important that this prediction can be performed substantially faster than real time to enable additional downstream processing in the future. The use of a five-frame sliding window for averaging of predictions does increase latency by the time required to acquire two frames beyond the frame of interest, thus increasing latency of an intraoperative application. However, this recursive averaging could be removed if decreased latency were desired while still maintaining excellent classification performance. In either scenario, the lightweight nature and the speed of inference of our selected model will allow for the implementation of more complex analyses on top of our current approach.

We experimented with many complex models; however, our simpler architecture consisting of only a DenseNet169 model and a dense NN layer performed best out of all those considered. Although we investigated many different architectures, the space of CNNs is very large, and it is possible that an alternative CNN network could yield greater performance. One CNN architecture we considered, but did not implement, was NasNet.[8] This architecture was too large for the 2-GPU setup we used and felt to be a reasonable reflection of standard hardware. It is possible that NasNet could improve performance by helping to optimize the wiring of our CNN as opposed to using a generic DenseNet model. This model warrants further investigation but highlights the tradeoffs of space and time described above.

The results also suggest that using RNNs for smoothing predictions does not yield significant improvements when training on the BigCat dataset. This is backed by our data, as the validation F1 scores do not significantly increase or decrease when using the LSTM on top of the DenseNet169. This could be because we trained our model on all frames in our training dataset, where many previous attempts sampled frames at a lower rate (e.g., 6 frames per second).[2] By sampling at a lower frame rate, it is possible that the loss of data requires a smoothing technique to ensure predictions are not erratic.

One reason that our simple model may have outperformed more complex architectures in previous works is the use of our dataset, BigCat. Our findings suggest that the large amount of annotated data in our BigCat dataset allows us to achieve exceptional performance with respect to identifying surgery tools in cataract surgery videos. Our initial models used for validation were all trained on 640,000 video frames, which amounts to around 28% of our dataset. We saw improvements over these models in our validation and testing performance when instead training on the full dataset. Additionally, our final model was trained on 2,282,382 video frames. DResSys was trained on only around 82,000 video frames.[2] The ability of our lightweight model to outperform DResSys is thus likely related to the size and quality of the BigCat dataset used to train our model.

Limitations of this study include the use of a testing dataset gathered from the same institution as the training and validation datasets. The absence of a comparable public dataset for external testing is a limitation of the current study. The publicly available Cataract 101 dataset, for example, does not contain instrument presence annotations, and has only surgical phase annotations. While the data augmentation performed on BigCat should allow for some invariance to scale and orientation, it will be valuable to assess performance on external datasets in the future. As instruments can have very different appearances (such as an irrigation-aspiration handpiece with polymer tip vs. silicone tip), true generalizability requires examples of all potential representations of a given instrument type, which will pose an ongoing data collection challenge moving forward.

Future work will involve the development of models to assess the actions of the instruments identified by the models presented here. In addition, we will look to expand BigCat to include complex cataract surgeries. This will enable future models to identify more rare surgical tools such as iris expansion devices, capsular hooks, and capsular tension rings and further improve their generalizability.

Acknowledgments

Supported by GME Innovations Fund (N.N., B.T.), The Doctors Company Foundation (N.N., B.T.), and NIH K12EY022299 (N.N.).

Disclosure: **N. Matton,** None; **A. Qalieh,** None; **Y. Zhang,** None; **A. Annadanam,** None; **A. Thibodeau,** None; **T. Li,** None; **A. Shankar,** None; **S. Armenti,** None; **S.I. Mian,** None; **B. Tannen,** None; **N. Nallasamy,** None

References

1. Lundström M, Behndig A, Kugelberg M, Montan P, Stenevi U, Thorburn W. Decreasing rate of capsule complications in cataract surgery: Eight-year study of incidence, risk factors, and data validity by the Swedish National Cataract Register. *J Cataract Refract Surg*. 2011;37:1762–1767.
2. al Hajj H, Lamard M, Charriere K, Cochener B, Quellec G. Surgical tool detection in cataract surgery videos through multi-image fusion inside a convolutional neural network. *Annu Int Conf IEEE Eng Med Biol Soc*. 2017;2017:2002–2005.
3. al Hajj H, Lamard M, Conze PH, et al. CATARACTS: Challenge on automatic tool annotation for cataRACT surgery. *Medical Image Analysis*. 2019;52:24–41.
4. Lewis DD, Gale WA, Croft IWB, van Rijsbergen CJ. A Sequential Algorithm for Training Text Classifiers. In: *SIGIR'94*. London: Springer; 1994: 3–12.
5. Sokolova M, Lapalme G. A systematic analysis of performance measures for classification tasks. *Inf Process Manag*. 2009;45:427–437.
6. Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. *J Big Data*. 2019;6(1):1–48.
7. Iandola F, Moskewicz M, Karayev S, Girshick R, Darrell T, Keutzer K. DenseNet: implementing efficient convnet descriptor pyramids. arXiv preprint arXiv:1404.1869. 2014 Apr 7.
8. Szegedy C, Ioffe S, Vanhoucke V, Alemi A. Inception-v4, inception-resnet and the impact of residual connections on learning. *31st AAAI Conference on Artificial Intelligence*. 2017 Feb 12.
9. Lobo JM, Jiménez-Valverde A, Real R. AUC: a misleading measure of the performance of predictive distribution models. *Global Ecology and Biogeography*. 2008;17:145–151.
10. Marzban C. The ROC curve and the area under it as performance measures. *Weather and Forecasting*. 2004;19:1106–1114.
11. Sorensen AT. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. *Biol Skar*. 1948;5:1–34.
12. Dice LR. Measures of the amount of ecologic association between species. *Ecology*. 1945;26:297–302.
13. Zoph B, Vasudevan V, Shlens J, Le QV. Learning transferable architectures for scalable image recognition. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2018:8697–8710.
14. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2016:770–778.
15. Quellec G, Lamard M, Cochener B, Cazuguel G. Real-time segmentation and recognition of surgical tasks in cataract surgery videos. *IEEE Trans Med Imaging*. 2014;33:2352–2360.
16. Schoeffmann K, Taschwer M, Sarny Klinikum Klagenfurt S, et al. Cataract-101-Video Dataset of 101 Cataract Surgeries. In: *Proceedings of the 9th ACM Multimedia Systems Conference*. 2018 Jun 12:421–425, doi:10.1145/3204949.
17. Yu F, Croso GS, Kim TS, et al. Assessment of automated identification of phases in videos of cataract surgery using machine learning and deep learning techniques. *JAMA Network Open*. 2019;2(4):e191860–e191860.
18. Zang D, Bian GB, Wang Y, Li Z. An Extremely Fast and Precise Convolutional Neural Network for Recognition and Localization of Cataract Surgical Tools. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Cham: Springer; 2019 Oct 13:56–64.
19. Morita S, Tabuchi H, Masumoto H, Yamauchi T, Kamiura N. Real-time extraction of important surgical phases in cataract surgery videos. *Sci Rep*. 2019;9(1):1–8.