CS230

Painting2Auction: Art Price Prediction with a Siamese CNN and LSTM

Tom Worth
Stanford University
tworth@stanford.edu

Abstract

Pre-auction predictions of painting price are central to the art auction process. These predictions have always been made by a set of highly-skilled individuals with extensive training, but these experts' time is scarce, so any degree of automation would be of great use. Deep learning has not yet been able to produce image-to-price predictions with any reasonable degree of accuracy, so machine learning has not had a meaningful impact on the art industry. I develop two new models for image-to-price prediction. Each of these incorporates not only the RGB data of the input image, but also recent market data, including prices and images of recently-auctioned paintings. My models use a Siamese CNN to evaluate similarity between the input painting and recently-auctioned paintings, and then use k-nearest-neighbors or an LSTM to predict the input painting's price from these similarity scores and their corresponding prices. Both models improve significantly on the performance of a naive CNN model, and the LSTM predicts test set prices with a Root Mean Squared Error of $279,079, compared to the naive CNN's RMSE of $854,743.

1 Introduction

Art price predictions are important for both auction houses and artists: they help the houses decide which pieces to show at exhibition and which pieces to group together in the same auction, and they help the artists' planning and decision-making in the interval between the production of a work and its sale. Artists would also benefit from automated predictions because automation would remove biases that human evaluators do not or cannot avoid, such as predicting higher prices for paintings by artists who have associations with presitigious institutions (Bailey [2020], Aubry et al. [2019]). These paintings then fetch higher prices as a sort of self-fulfilling prophecy, and the elitism in the art world is self-reinforced. The artist-agnostic evaluation made possible by automation would represent a step toward a more open and equitable art market. Finally, from the perspective of the market as a whole, more widely-available price predictions would increase liquidity: increased information would lead to more informed risk-taking and therefore greater productivity overall, both for buyers and sellers (Bailey [2020]). Automation of prediction would circumvent the bottleneck of valuation specialists' limited capacity, dramatically improving the accessibility of this crucial information, to everyone's benefit.

The problem, then, is to produce a price prediction which is as close as possible to the eventually-realized price, given a painting and the market context in which it is to be auctioned. In algorithm terms, the inputs are an image of the painting to be auctioned, images of other recently-auctioned paintings, those other paintings' time of auction, and their realized prices. The output is a dollar value.

CS230: Deep Learning, Fall 2020, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

My approach to this problem combines powerful deep learning techniques from computer vision and sequence modeling, the Siamese CNN and the LSTM, to handle this mix of image and sequential data. To my knowledge, this is the first time that the painting price prediction problem has been addressed in a way that both accounts for market context and excludes artist bias, and the result is an improvement on the accuracy and accessibility of previous approaches.

2 Related work

Past attempts at price prediction have generally varied from my approach in one of two ways. In some cases they have not taken market context into account, trying instead to go from an image directly to a price (Verge and Singal [2019], Ayub et al. [2017]). This framing of the problem fails to account for the fact that a given painting may fetch dramatically different prices at different points in time, which at least partially explains why this sort of model performs so poorly, failing to classify paintings into discretized price categories any better than chance (Ayub et al. [2017]).

A different sort of alternative approach brings much more data to bear on the problem, such as biographical information on the artist and text representing expert opinions or media articles on the piece. All of this information is brought together using NLP and a random forest model (Aubry et al. [2019], Tepper [2020]). This approach yields much more accurate predictions (Aubry et al. [2019]), but at the cost of objectivity and accessibility: factoring in expert opinions reintroduces potentially elitist biases, and besides, if these expert opinions are available, then the automated prediction is redundant anyway. This approach thus compromises a large part of the motivation to automate in the first place.

3 Dataset and Features

I made use of two datasets: one that has images with artist labels only (no price data), for the training of my Siamese classification network that was price-agnostic; and one that has painting images labeled with price realized and date of auction.

The image-to-artist dataset came from Kaggle (Kaggle [2015]). It has 7,943 images of paintings, of varying shapes and resolutions, covering a broad range of styles and time periods. To construct a training set for my Siamese CNN, I first removed any paintings by artists with fewer than 30 works in the dataset (following from Viswanathan [2017]). For each artist remaining, I formed 30 positive training pairs (pairing thirty paintings by that artist with another random painting by the same artist) and thirty negative training pairs (pairing thirty paintings by that artist with a random painting by a different artist), resulting in a balanced dataset of 6,708 pairs of paintings, again taking after Viswanathan [2017]. Each pair was labeled "1" for "same artist" or "0" for "different artist." All the images were originally JPEG files; I resized and reformatted them all to a $224 \times 224 \times 3$ array with pixel values normalized to values between 0 and 1.

A positive pair (above) and a negative pair (left)

The image-price-time dataset was scraped from the website of Phillips (phi), a major auction house. It contains a datapoint for each of 23,825 sales, each datapoint including an image, a price (converted to USD, ranging from $13 to $63,362,500), and a time of auction (Day/Month/Year, between October 2006 and October 2020). I also resized and reformatted each of these JPEG images to a $224 \times 224 \times 3$ array with normalized pixel values. See Appendix 7.1 for an example datapoint.

4 Methods

My model architectures have two distinct parts. The first part is a Siamese CNN which effectively produces a distance vector for any two input images; this part is the same for Model 1 and Model 2. The second part is a price prediction model: k-nearest-neighbors for Model 1, and an LSTM for Model 2. All models were created with Keras on TensorFlow (Chollet et al. [2015], Abadi et al. [2015]).

4.1 Siamese CNN

My first set of experiments evaluated the efficacy of several different CNN architectures for evaluating the similarity of two paintings, judged by their accuracy on the binary classification task ("same artist" or "different artist") with the Kaggle dataset. Each CNN is individually incorporated into a Siamese Neural Network, which passes the two paired images through identical copies of the CNN to reduce each image to a 128-entry vector encoding. These encodings are then subtracted, and the resulting difference vector is put through two densely-connected layers to reduce it to a single value. The last layer has a sigmoid activation function, and the prediction of true or false is decided upon by whether this final activation is closer to one or zero. The prediction is compared to the label, and the loss is evaluated with the binary cross-entropy loss function. My model uses Adam optimization to train all trainable weights in the model. The same updates are always made to the two identical CNNs, so that they remain identical across all iterations.

I evaluated a multilayer perceptron as a baseline; a vanilla CNN with two convolutional layers; a deeper vanilla CNN; and three larger models trained on ImageNet (ResNet-50, Inception v3, and EfficientNet b7). The purpose of these tests was to pick the optimal CNN architecture to incorporate in my two price prediction models.

4.2 Model 1: Price Prediction with K-Nearest-Neighbors

This model predicts the price of an input painting by taking the median of the price values of the k paintings closest to the input out of a pool of the n most-recently-auctioned paintings, where k and n are hyperparameters and "closest" is judged with the Siamese CNN. More specifically, the Siamese CNN returns the probability that each of the last n paintings was painted by the same artist as the input painting, and the top k paintings with respect to this probability are taken to be the k nearest neighbors. The median of these paintings' prices is returned as the prediction for the input painting's price, not the mean as is more common with k-nearest-neighbors, because outlier painting prices skew the mean too greatly. Accuracy is evaluated using Mean Absolute Percentage Error, for its ease of interpretation.

4.3 Model 2: Price Prediction with LSTM

The second model connects the Siamese CNN directly to a simple LSTM, allowing parameter updating to flow through the LSTM all the way to the convolutional layers of the image comparison network. This allows the painting comparison stage to be further fine-tuned to the problem of price prediction, rather than freezing it with training only from the binary classification task.

The Siamese CNN architecture with the best performance on the binary classification task is used to compute a 128-entry feature "distance" vector for each of the n most-recently-auctioned paintings, where n is a fixed number of sales, treated as a hyperparameter. Each of these feature vectors comes from using the input painting image (whose price is to be predicted) and one of the recently-auctioned painting images as the inputs to the Siamese CNN, and extracting the feature vector from the last layer of the network just before it is collapsed into a probability (which would

have been the output in the original implementation). The price of the recently-auctioned image is appended onto the end of this feature vector, and the resultant 129-entry vector is one timestep in the sequence which will constitute the LSTM's input.

The LSTM itself is a simple many-to-one architecture, with only one recurrent layer, with 10 hidden units and ReLU activation. I use such a simple model because of memory constraints, but performance shows that this number of units and layers is sufficient to achieve good prediction accuracy.

See Appendix 7.2 for a diagram summarizing the LSTM architecture.

I used Mean Squared Error loss and Adam optimization, and tuned learning rate and input sequence length as hyperparameters. Validation and test accuracy were evaluated with Root Mean Squared Error and Mean Absolute Percentage Error, for interpretability.

5 Results

5.1 Baseline Models

I implemented a CNN which predicts price straight from an image, without looking at all at past auction data, for a baseline. This is essentially a reimplementation of Verge and Singal [2019], but without the data restriction which accounted for their improved accuracy. This model achieved an RMSE of $854,743 (MAPE = 1,841).

For context, I also made a "predictor" which simply returns the price of a random other painting from the dataset. Across 1,000,000 of these "predictions," this resulted in an RMSE of $1,570,903 (MAPE = 2,351). It makes sense that this is worse than, but in the same ballpark as, the performance of the naive CNN.

5.2 Siamese CNN Results

The following table summarizes the results of my evaluation of different CNN architectures in the Siamese network for the binary classification task. I used a $70/15/15$ split on the dataset since it was small, so the training set had 4,695 pairs of images, and the validation and test sets had 1,006 pairs each. The values in the table are after 8 training epochs with mini-batch size 40. See Appendix 7.3 for learning curves.

CNN Architecture	Val Accuracy (%)
Multilayer Perceptron	52
Vanilla CNN	86
Deeper Vanilla CNN	72
ResNet-50	53
Inception v3	65
EfficientNet b7	75

Further training under the same conditions only harmed validation accuracy. For each of the three large CNNs, pretrained weights up to this point have been frozen; fine-tuning the models with pretrained weights set to "trainable" and a smaller learning weight led to slightly improved accuracy, as expected, but did not cause any of the models to outperform the Vanilla CNN on the validation set (the best transfer learning performance was by the Inception net, which achieved 82% val accuracy after 8 additional epochs, which is still worse than the Vanilla's 86%).

To summarize, the Siamese CNNs show success in extracting the most relevant features of paintings to determine similarity in style and content, and the best performance is achieved by the Vanilla CNN (see Appendix 7.4 for a diagram of this model architecture), which achieved an accuracy of 86%. This high rate is promising for its ability to translate to application in the second part of my algorithm, where it will be used to identify which features of a painting make it "like" another painting, in a way that is meaningful for price prediction.

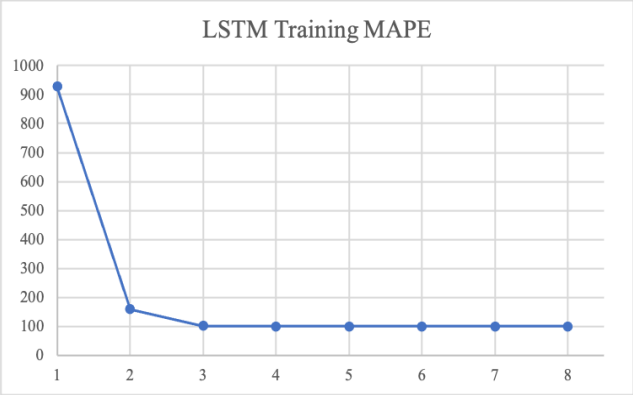
5.3 Price Prediction

5.3.1 K-Nearest-Neighbors Results

Using the Siamese architecture with the Vanilla CNN as the distance evaluator for the K-Nearest-Neighbors approach, the model achieved an MAPE of 227. This was with hyperparameters k and n set to 10 and 200 respectively; see Appendix 7.5 for a table of results for a range of values of k and n. This is already a significant improvement on the baselines. See Appendix 7.6 for visualized examples of "nearest neighbors" produced by this model.

5.3.2 LSTM Results

The LSTM (again with the Vanilla CNN incorporated as the CNN for its Siamese Net stage) performed even better, achieving a test set performance of MAPE 100.08 (RMSE = $279,079). The hyperparameters on that model were sequence length 20 and learning rate .001. See Appendix 7.7 for validation set performance across a range of these hyperparameters. The following figures show the learning curve of test MAPE for this last model and an example of a well-predicted price from the trained LSTM.



6 Discussion and Future Work

I introduced two new approaches to art price prediction, and their performance represents a marked improvement over existing deep learning methods for painting valuation. They demonstrated a great ability to make sense of a combination of visual input and recent auction data, leveraging both types of input to produce high prediction accuracy, all without introducing any artist-related bias.

I first evaluated different architectures for a neural network that evaluated paintings' similarity in a way that was both meaningful and relevant to price prediction. The K-Nearest-Neighbors model then made use of a simple statistical technique to harness the values produced by this network, attaining a Mean Absolute Percentage Error of 227. I then implemented an LSTM which made use of the same comparison network but accounted for the time series aspect of the data in a more nuanced way, producing a final test set MAPE of only 100.08. Evaluated against the baseline performance of a naive regression CNN, which was only able to reach an MAPE of 1,841, the two new models show significant progress toward an automated painting evaluation process with real-world applicability.

Future work with models like this should focus mainly on moving to bigger data, since my datasets were relatively small by the standards of deep learning on image inputs, and I was somewhat inhibited by memory constraints in evaluating hyperparameters for the LSTM. The optimal sequence length of 20 may not generalize to larger datasets. Additionally, future work could also involve developing a more efficient storage of historical auction data, for example storing feature vectors for paintings rather than the images themselves. This could also help with transitioning to bigger training datasets to further improve on performance.

7 Appendix

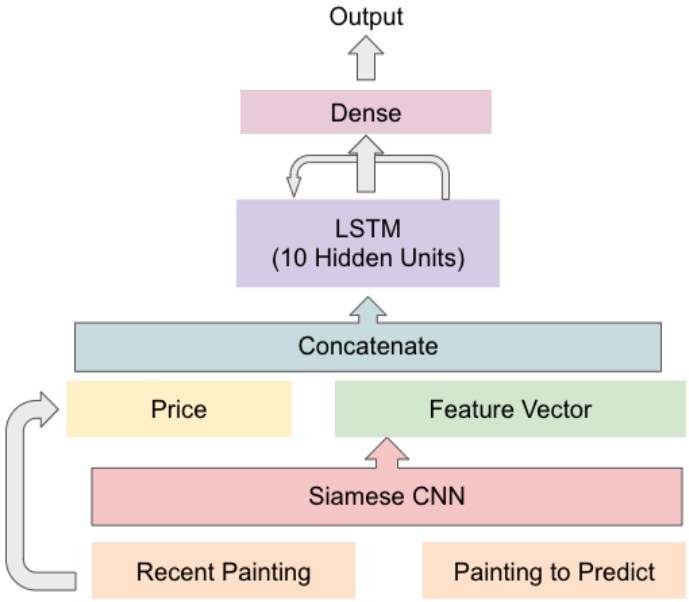
7.1 Phillips Dataset Example

Tom Wesselmann,
Study for Still Life with Fruit, Daisies and Monica.

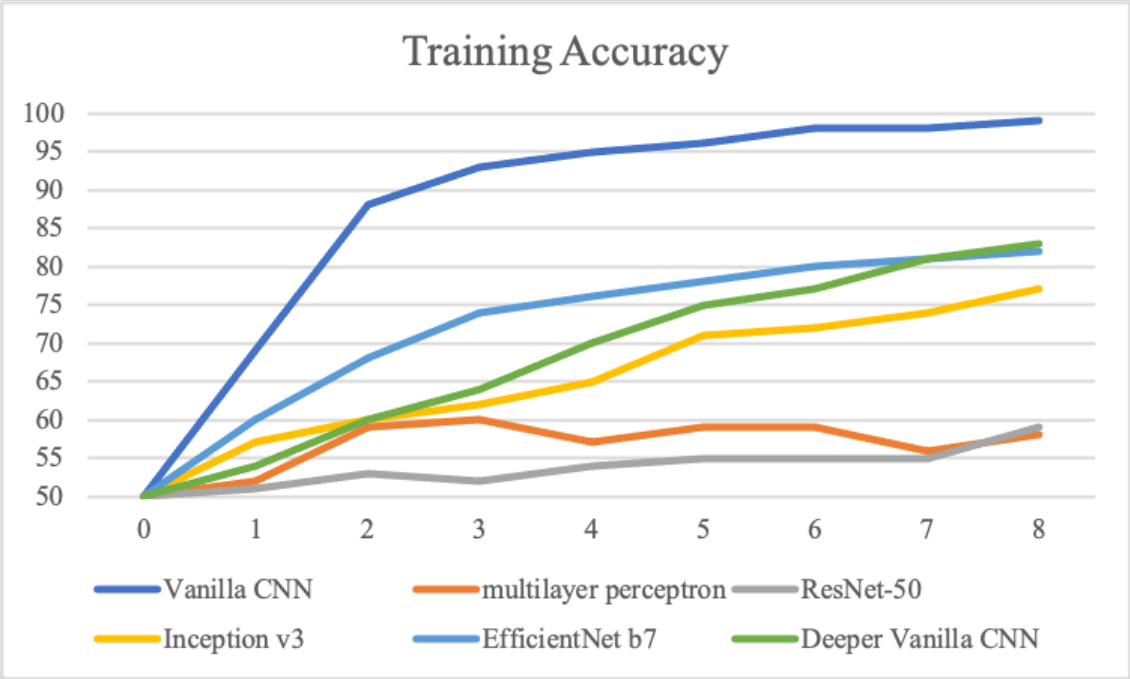
Auctioned 10/21/2020.

Price realized:
$106,470.00

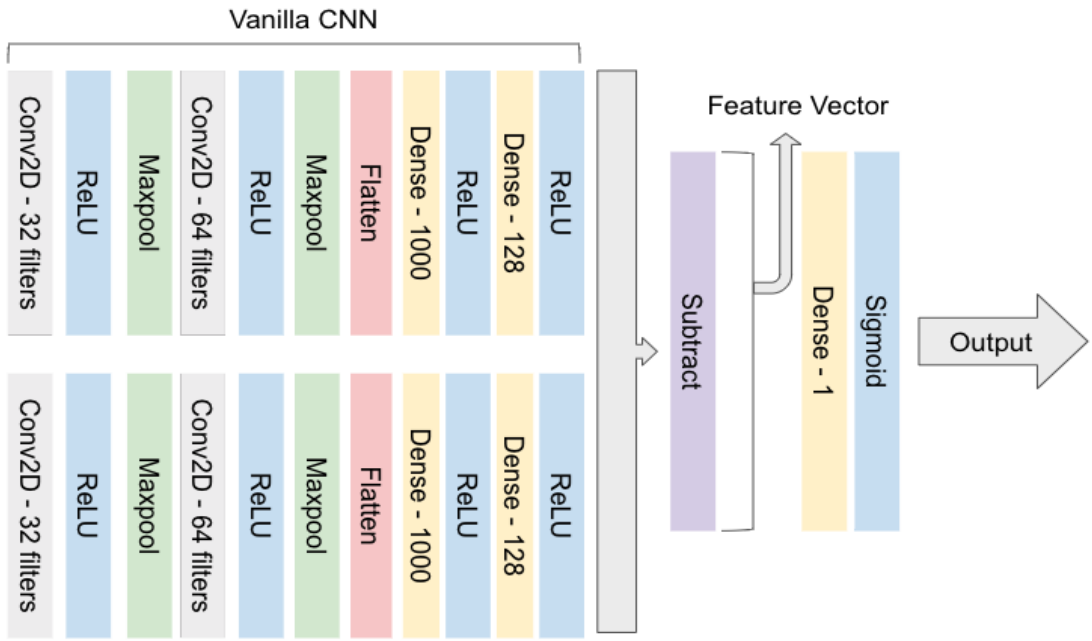
7.2 LSTM Architecture with Siamese CNN



7.3 Siamese Net learning across different CNN architectures



7.4 Siamese Net architecture with Vanilla CNN



7.5 Nearest Neighbors Hyperparameters

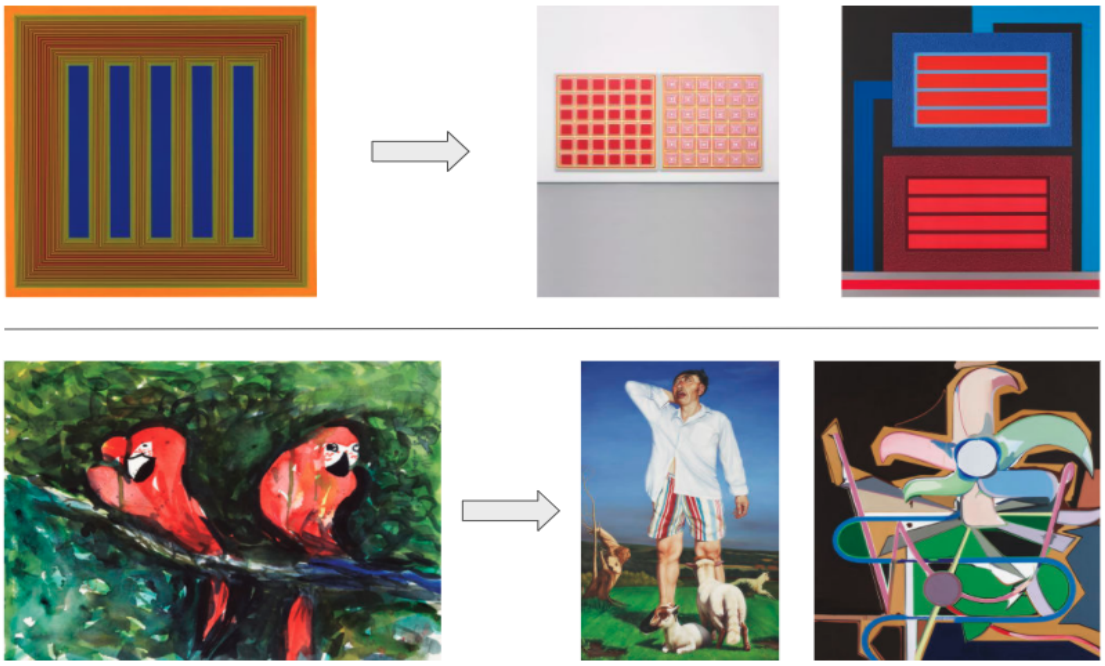
k	n	MAPE
10	100	983
30	100	946
50	100	589
70	100	2551
90	100	1648

k	n	MAPE
10	200	227
30	200	4891
50	200	5276
70	200	1513
90	200	4822

k	n	MAPE
10	400	662
30	400	653
50	400	592
70	400	722
90	400	821

7.6 Nearest Neighbors Visualization

The following figure shows two examples of two nearest neighbors for a given input, as evaluated by the Siamese net with the Vanilla CNN. It is evident that the Siamese CNN is picking up on relevant features of paintings, such as recurrent patterns, similarity in texture, similarity in content, and even similarity in abstract forms, as shown by the vaguely bird-like shapes in both of the images that were identified as similar to the parrots.



7.7 LSTM Hyperparameter Tuning

n	Learning Rate	Val MAPE
10	.001	180
20	.0001	165
20	.001	98.54
20	.01	285
50	.001	319

References

Jason Bailey. Can machine learning predict the price of art at auction? *Harvard Data Science Review*, Apr 2020. doi: 10.1162/99608f92.7f90ce96.

Mathieu Aubry, Roman Kraeussl, Gustavo Manso, and Christophe Spaenjers. Machines and masterpieces: Predicting prices in the art auction market. *SSRN Electronic Journal*, Mar 2019. doi: 10.2139/ssrn.3347175.

Alexander Verge and Ishaan Singal. State-of-the-art: End to end deep learning for art appraisal. 2019.

Rafi Ayub, Cedric Orban, and Vidush Mukund. Art appraisal using convolutional neural networks. Dec 2017.

Nona Tepper. How artnome uses machine learning to predict the price of an artistic masterpiece, Apr 2020. URL `https://builtin.com/machine-learning/artnome-boston-machine-learning`.

Kaggle. Painter by numbers. 2015. URL `https://www.kaggle.com/c/painter-by-numbers/data`.

Nitin Viswanathan. Artist identification with convolutional neural networks. 2017. URL `http://cs231n.stanford.edu/reports/2017/pdfs/406.pdf`.

Auction results. URL `https://www.phillips.com/auctions/past/filter/Departments=Contemporary/sort/newest`.

Francois Chollet et al. Keras, 2015. URL `https://github.com/fchollet/keras`.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.