Pie Chart or Pizza: Identifying Chart Types and Their Virality on Twitter

Pavlos Vougiouklis,[1] **Leslie Carr,**[1] **Elena Simperl**[2]
School of Electronics and Computer Science
University of Southampton
Southampton, United Kingdom
[1]{pv1e13, lac}@ecs.soton.ac.uk, [2]E.Simperl@soton.ac.uk

Abstract

We aim to understand how data, rendered visually as charts or infographics, "travels" on social media. To do so we propose a neural network architecture that is trained to distinguish among different types of charts, for instance line graphs or scatter plots, and predict how much they will be shared. This poses significant challenges because of the varying format and quality of the charts that are posted, and the limitations in existing training data. To start with, our proposed system outperforms related work in chart type classification on the ReVision corpus. Furthermore, we use crowdsourcing to build a new corpus, more suitable to our aims, consisting of chart images shared by data journalists on Twitter. We evaluate our system on the second corpus with respect to both chart identification and virality prediction, with promising results.

1 Introduction

We live in a world full of data, in which charts are routinely used to communicate complex insights more effectively than spreadsheets or reports (Gray, Chambers, and Bounegru 2012; Savva et al. 2011; Jung et al. 2017). Twitter is no exception—tens of thousands of data visualisations on virtually any topic are shared every day. Specialist accounts such as "Information is Beautiful", based on the eponymous visual design book by David McCandless, have reached more than 100 thousand of followers. Media outlets have set up dedicated accounts (e.g. nytgraphics and ReutersGraphics) to promote their data storytelling work, which focuses on disseminating information and analysis using charts. Brands have discovered infographics and other visual renderings of data as a way to boost traffic and reach a larger audience—for instance, to promote Narcos, a show that tells the story of Pablo Escobar, Netflix launched a data story[1] that talks about the economy of Columbian cocaine trade in a socially engaging way.

Our aim is to understand how data rendered visually as charts or infographics "travels" on Twitter. Studies in how information spreads on social media have shown that the exposure a post will get is influenced by many factors, including topic, presentation, timing and social status of the author (Guerini, Staiano, and Albanese 2013; Khosla, Das Sarma, and Hamid 2014; Deza and Parikh 2015). Data visualisations should certainly be no different, though research in this space is in its beginnings. To this end, we propose a neural network architecture that is trained to identify whether a post includes a chart, distinguish among different types of charts, for instance line graphs or scatter plots, and predict how much they will be shared.

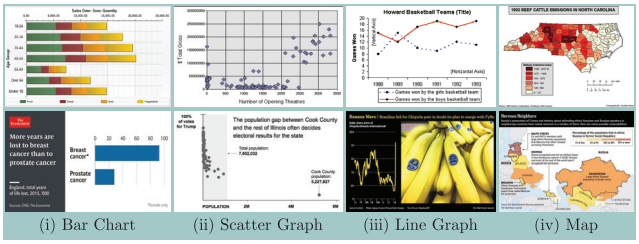


Figure 1: Examples of different chart types in ReVision (top line) and in Twitter posts by some major news agencies (bottom line). The ones posted on Twitter tend to be augmented with additional chunks of text and non-chart-related images.

Building such as system poses several challenges, mostly because of the varying format and quality of the charts, and the limited training data available. Unlike general-purpose visual recognition (Krizhevsky, Sutskever, and Hinton 2012; Simonyan and Zisserman 2015), identifying and classifying charts is much less explored, and only in idealised scenarios (Prasad et al. 2007; Savva et al. 2011; Jung et al. 2017). By comparison, the data visualisations that are shared on social media often include additional elements, such as text and chunks of images. Figure 1 makes this explicit by juxtaposing examples from existing benchmark datasets used in prior work (i.e. ReVision (Savva et al. 2011)) and from posts published by major news agencies on Twitter. We hypothesised that models that were trained on idealised images do not generalise well to charts from a Twitter feed.

To test this hypothesis, we adapted and trained from scratch a state-of-the-art Convolutional Neural Network

[1]https://twitter.com/hashtag/cokenomics. Accessed 27 Aug. 2019.

(ConvNet) (Simonyan and Zisserman 2015) on the ReVision corpus. We augmented this dataset with a set of non-chart images. As a result, in comparison to existing chart-type identification approaches (Prasad et al. 2007; Savva et al. 2011; Jung et al. 2017), our learned system was not only able to classify an image according to its chart type, but also to identify whether the image displays a chart in the first place. The system achieved a test accuracy score of 93.61% on this 12-class classification task (11 chart types plus an independent non-chart class). It also outperformed the competing systems by Savva et al. and Jung et al. on the 10 chart types from the ReVision dataset.

To train a model suitable for our aims, we used crowdsourcing to build a new corpus, consisting of 3000 image tweets that have been posted by the Twitter accounts of some major news agencies. Each image in the corpus has been labelled with the relevant chart type (or types, where applicable). We make the resulting corpus publicly available at: https://github.com/pvougiou/Pie-Chart-or-Pizza. Our hypothesis was confirmed—the performance of the pre-trained ConvNet architecture was considerably lower on this more challenging corpus. To improve it, we adapted a training strategy that enabled our system to achieve an accuracy of around 86%.

We then used the learned visual features in order to predict the diffusion of chart-driven information on Twitter. Since a user is exposed[2] to both their followers' retweets and favourites (or *likes* since 2015), we modelled the *virality potential* of a tweet as a function of retweet and favourite counts. Our final system jointly learns to make a prediction for both these counts given a chart post on Twitter. It consists of: (i) a ConvNet that extracts features from the chart image; (ii) a bidirectional architecture with Gated Recurrent Units (GRUs) (Cho et al. 2014) that processes the accompanying text; and (iii) a feed-forward architecture that expects a set of features that describe its author. We experimented with alterations of this model using different input signals (e.g. with or without the author- and the image-related features) in order to determine their effects on the prediction. Our approach was inspired by recent work by Zhao et al., who used a multi-modal neural architecture on the binary task of retweet prediction. However, unlike Zhao et al., we did not use the pre-trained ConvNet by Simonyan and Zisserman. Since our task focuses on charts, we trained and fine-tuned ours from scratch. In addition, our system relied on computationally less expensive author-related features that do not model each user's past retweet behaviour and following relations. Despite the simpler design, the experiments confirmed that their inclusion results in substantial performance improvements. The main contributions of this paper are as follows:

- A *ConvNet architecture for chart classification*, which outperforms other competing systems on the ReVision benchmark, while also being able to exclude images that do not contain charts.

[2]See instructions at: https://help.twitter.com/en/managing-your-account/understanding-the-notifications-timeline. Accessed 27 Aug. 2019.

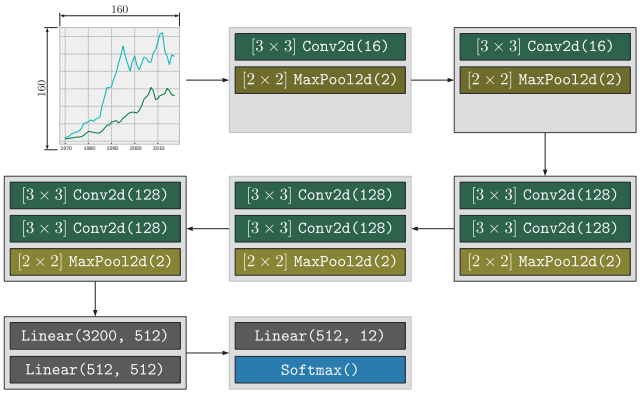


Figure 2: The architecture of our ConvNet

- A new *dataset of real-world data visualisations from Twitter*. Using this dataset, we show the problem of applying approaches trained on idealised corpora to data collected in the wild. To overcome it, we devise a *training strategy* that improves the accuracy of our end-system by more than 15%.
- A *multi-modal neural architecture* that jointly learns to predict the number of times a chart post will be retweeted and liked. We show that coupling textual and author-related features with our learned visual features results in more accurate predictions.

2 Background

Our work synthesises two strands of research, chart identification and virality prediction of images on social media.

2.1 Classifying Data Visualisations

The classification of general-purpose images has been thoroughly investigated in the literature. There has been a substantial amount of work that proposes various adaptations of ConvNets for large-scale visual recognition tasks, with promising results (Krizhevsky, Sutskever, and Hinton 2012; Simonyan and Zisserman 2015). Nonetheless, the classification of chart images according to their type has only been sporadically explored. Shao and Futrelle and Huang and Tan and classified vector images across 5 (Shao and Futrelle 2006) and 4 (Huang and Tan 2007) different chart types, respectively by first extracting high-level shapes from them. Prasad et al. used features based on curve saliency, local segmentation, Histogram Oriented Gradients and Scale Invariant Feature Transform to represent a given chart image (Prasad et al. 2007). They trained a multi-class SVM with only the images whose features were found to be the most discriminative according to the Pyramid Match algorithm.

Savva et al. introduced the ReVision corpus, and expanded the classification task to 10 chart types (Savva et al. 2011). They proposed a set of visual and text features that were computed by extracting visual patches and regions of text from a given chart image. They trained a multi-class SVM, achieving an average accuracy of 80%. More recently, Jung et al. employed an out-of-the-box ConvNet

(Szegedy et al. 2015) on ReVision achieving an accuracy of $76.7\% - 85\%$ (Jung et al. 2017). In comparison to these works, we introduced one additional chart type and augmented the training data with a non-chart class, which enabled our system to exclude irrelevant images (for instance the picture of a pizza vs a pie chart). While our architecture was trained on a more challenging task and used less than half of the parameters from (Jung et al. 2017), it outperformed both the above approaches when tested on ReVision.

2.2 Predicting Image Virality

Understanding the diffusion of visual content on social media is critical for many domains—from marketing to fake news. Several prior works have tried to identify what an image is about and to extract the most relevant features to predict virality (Can, Oktay, and Manmatha 2013; Khosla, Das Sarma, and Hamid 2014; Deza and Parikh 2015; Zhao et al. 2018). This is modelled according to some *exposure* metric (e.g. number of upvotes and downvotes (Deza and Parikh 2015) or number of views (Khosla, Das Sarma, and Hamid 2014)), depending on the social media platform. The task can be designed as binary classification by ranking images according to the metric and allocating the ones at the top and bottom of the scale to their respective classes (Deza and Parikh 2015); or as regression by seeking to directly estimate the metric (Can, Oktay, and Manmatha 2013; Khosla, Das Sarma, and Hamid 2014). We opted for the latter and built a system that jointly learns to predict counts of retweets and likes of a chart message. To the best of our knowledge, this work is the first attempt to estimate the virality of visualisations in the context of social media.

3 Our System

We first present the architecture of the ConvNet that identifies whether an image displays a chart, and in case it does, its exact chart type. We used a pre-trained version of this architecture as part of a multi-modal neural architecture that jointly learns to predict the number of times a chart tweet will be retweeted and liked—see Figure 3 for an overview.

3.1 Chart Images Identification

We adapted the VGGNet architecture proposed by Simonyan and Zisserman to the requirements of our chart identification problem. This architecture has been originally trained and evaluated on the ILSVRC-2012 task that consists of 1.4M images distributed across 1000 classes. Our scenario was different—we had considerably fewer classes (12 classes versus 1000), but also much less training data (the ReVision dataset, see below). For these reasons, we needed an implementation that is computationally more efficient.

During both training and testing the input to our ConvNet was a fixed-size 160×160 RGB image. Similarly to VGGNet, the network used a stack of convolutional layers, all of which with a receptive field of 3×3. The convolution stride was set to 1. The spatial dimension of convolutions was preserved by zero-padding their input volumes with 1 pixel on each side. We performed max pooling over 2×2 pixel windows with a stride of 2. The stack of convolutional layers was followed by three fully-connected layers, where the latter used a softmax function to predict the class to which the input image belonged. All convolutional and fully-connected layers used the ReLU as non-linear activation function.

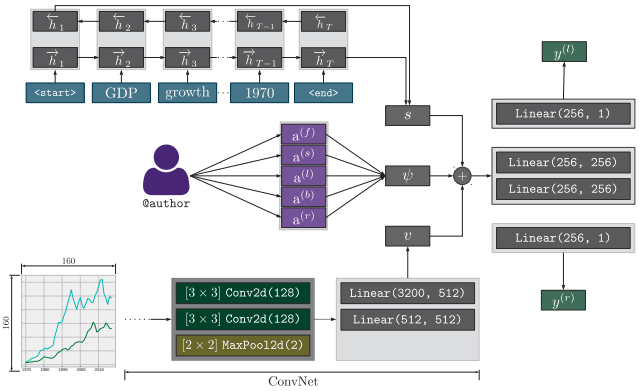


Figure 3: Our multi-modal neural architecture

Figure 2 displays the architecture of our ConvNet. In principle, the architecture is similar to the "A" configuration of VGGNet (Simonyan and Zisserman 2015). Nonetheless, the number of feature maps per convolutional layer and the dimensionality of the hidden states in the fully-connected layers is much lower in our case. In addition, the fact that the fixed resolution of the input images is lower than the one expected in VGGNet results in smaller feature maps output by the last max pooling layer. Deep neural networks are susceptible to overfitting, especially in the case of small training corpora such as ours (Cogswell et al. 2015). Consequently, our goal was to reduce the number of our system's learnable parameters to limit its overfitting tendencies (Cogswell et al. 2015). Our ConvNet consisted of 2482k (excluding the final fully-connected layer) parameters, which is around 129M weights less than VGGNet's "A" configuration.

3.2 Virality Prediction

Let $e^{(a,m,\mathbf{x})}$ be a message posted on Twitter by the user a, where m and $\mathbf{x}$ are the chart image and the snippet of text that accompany the message. Let also $x_1, \dots x_T$ be the words of which $\mathbf{x}$ consists s.t. $\mathbf{x} = (x_1, \dots, x_T)$. We built a model that predicts the number of times that $e^{(a,m,\mathbf{x})}$ would be retweeted, $y^{(r)} \in \mathbb{R}$, and liked, $y^{(l)} \in \mathbb{R}$. Our end-to-end architecture consists of: (i) a ConvNet that extracts visual features from the chart image m, (ii) a feed-forward architecture that processes the author's, a, characteristics, and (iii) a bidirectional GRU that processes the information in the text $\mathbf{x}$.

Processing the Chart Image We used a pre-trained version of the ConvNet presented in Section 3.1 to extract visual features from the chart image m from a tweet. Let $v \in \mathbb{R}^{512}$ be the output of this ConvNet (i.e. ConvNet$_\theta$), stripped of its last fully-connected and softmax layers. The vector representation of a given chart m, v, is computed by forward

propagating as follows: $v = \text{ConvNet}_\theta(m)$.

Processing the Text We used a bidirectional GRU to encode the information in $\mathbf{x}$. Let $\overrightarrow{h}_t^l, \overleftarrow{h}_t^l \in \mathbb{R}^m$ be the aggregated output of a hidden unit of the forward and backward pass respectively at timestep $t = 1 \dots T$ and layer depth $l = 1 \dots L$. The vectors at zero layer depth, $h_t^0 = \mathbf{W}_{\mathbf{x}\to\mathbf{h}} x_t$, represent the tokens, $x_1, \dots, x_T$, of $\mathbf{x}$ that are given to the network as input. The parameter matrix $\mathbf{W}_{\mathbf{x}\to\mathbf{h}}$ has dimensions $[|X|, m]$, where $|X|$ is the size of the input dictionary. We initialised this matrix using GloVe embeddings (Pennington, Socher, and Manning 2014) and allowed the network to fine-tune it during training. All the subsequent matrices have dimension $[m, m]$ unless stated otherwise. At each timestep t, $\overrightarrow{h}_t^l$ and $\overleftarrow{h}_t^l$ are computed as follows:

$$\overrightarrow{h}_t^l = \text{GRU}(\overrightarrow{h}_{t-1}^l, h_t^{l-1}), \tag{1}$$

$$\overleftarrow{h}_t^l = \text{GRU}(\overleftarrow{h}_{t-1}^l, h_t^{l-1}). \tag{2}$$

The context vector $h_t^l \in \mathbb{R}^{2m}$ that encapsulates the information from both the forward and backward pass at each layer l and timestep t is computed as $h_t^l = [\overrightarrow{h}_t^l; \overleftarrow{h}_t^l]$, where $[\dots;\dots]$ represents vector concatenation. Subsequently, the vector that encapsulates all the information from $\mathbf{x}$, is computed by aggregating the hidden states of the two passes at their last processing timestep (i.e. $t = T$ and $t = 1$ for the forward and backward pass respectively) of the topmost layer s.t. $s = [\overrightarrow{h}_T^L; \overleftarrow{h}_1^L]$.

Processing the Author We incorporated the author-related features of Can, Oktay, and Manmatha in our multi-modal architecture (Can, Oktay, and Manmatha 2013). The vector that represents the author who posted the message $e^{(a,m,\mathbf{x})}$ is computed as follows:

$$\psi = [a^{(f)}; a^{(s)}; a^{(l)}; a^{(b)}; a^{(r)}], \tag{3}$$

where $a^{(f)}, a^{(s)}, a^{(l)}, a^{(b)} \in \mathbb{R}$ are the total number of followers, posts, likes and friends (i.e. number of accounts that the author follows) respectively that a has and $a^{(r)} \in \mathbb{R}$ is the ratio of the number of followers to the number of friends. Following Can, Oktay, and Manmatha, we transformed the $a^{(f)}$, $a^{(s)}$, $a^{(l)}$ and $a^{(b)}$ values to logarithmic scale[3] before feeding them into our model (Can, Oktay, and Manmatha 2013).

Likes and Retweets Prediction After computing the s, v and ψ vector representations for the text, chart image and author respectively, the system projects the three modalities into a shared feature space. The multi-modal context vector c_e for $e^{(a,m,\mathbf{x})}$ is computed as:

$$c_e = \text{ReLU}\left(\mathbf{W}_\mathbf{a}\psi + \mathbf{W}_\mathbf{s}s + \mathbf{W}_\mathbf{v}v\right), \tag{4}$$

where $\mathbf{W}_\mathbf{a} : \mathbb{R}^5 \to \mathbb{R}^m$, $\mathbf{W}_\mathbf{s} : \mathbb{R}^{2m} \to \mathbb{R}^m$ and $\mathbf{W}_\mathbf{v} : \mathbb{R}^{512} \to \mathbb{R}^m$ are biased linear mappings. Our formulation is similar to Zhao et al.'s "multi-modal fusion layer" (Zhao et al. 2018). Zhao et al. computed the multi-modal context vector using solely the extracted visual and textual features. They subsequently ranked the preference of the author's followers towards this vector. The design of our user features enabled us to directly include them in the computation of the context vector. Furthermore, Zhao et al. used the hyperbolic tangent in their multi-modal fusion layer. However, in our experiments, this resulted in lower performance compared to the rectifier. After computing the context vector c_e, our architecture predicts the expected number of retweets and likes for $e^{(a,m,\mathbf{x})}$ as follows:

$$\tilde{y} = \text{ReLU}\left(\mathbf{W}_\mathbf{c}^{(\mathbf{II})}\text{ReLU}\left(\mathbf{W}_\mathbf{c}^{(\mathbf{I})}c_e\right)\right), \tag{5}$$

$$y^{(r)} = \mathbf{W}_\mathbf{y}^{(\mathbf{r})}\tilde{y} \quad \text{and} \quad y^{(l)} = \mathbf{W}_\mathbf{y}^{(\mathbf{l})}\tilde{y}, \tag{6}$$

where $\mathbf{W}_\mathbf{c}^{(\mathbf{I})}, \mathbf{W}_\mathbf{c}^{(\mathbf{II})} : \mathbb{R}^m \to \mathbb{R}^m$ and $\mathbf{W}_\mathbf{y}^{(\mathbf{r})}, \mathbf{W}_\mathbf{y}^{(\mathbf{l})} : \mathbb{R}^m \to \mathbb{R}$ are biased linear mappings.

Training We modelled virality prediction as a regression task. During training, our model aims to minimise the sum of the squared losses of the predicted retweet and like counts with respect to their target values, $y_t^{(r)}$ and $y_t^{(l)}$:

$$\text{cost} = \|y^{(r)} - y_t^{(r)}\|_2^2 + \|y^{(l)} - y_t^{(l)}\|_2^2. \tag{7}$$

We addressed the large variation in the number of retweets and likes of different chart tweets by computing the natural logarithm[3] of the target values before bootstrapping them in the loss function (Can, Oktay, and Manmatha 2013; Khosla, Das Sarma, and Hamid 2014).

4 Datasets

In this section, we present the three corpora that we used for training and evaluation: (i) ReVision+, (ii) DataTweet, and (iii) DataTweet+. The first is based on ReVision, which is provided by Savva et al.. The other two are collections of image tweets authored by data journalists and labelled using crowdsourcing. For the purpose of the experiments, all three corpora are randomly split into training, validation and test, with respective portions of 70%, 15% and 15% for ReVision+ and DataTweet, and 60%, 20% and 20% for DataTweet+. Since DataTweet+ is smaller than the other two corpora, we increased the portion of its validation and test set to guarantee more representative samples.

4.1 ReVision and ReVision+

The original ReVision corpus contains 2965 images of charts distributed across 15 categories. In addition to the 10 categories used by Savva et al. and Jung et al., we considered: (i) column charts as part of the "bar chart" class; (ii) box plots as an extra class; and (iii) non-chart images. Incorporating column charts as part of the bar chart class enabled us to increase the number of available images for this class. In preliminary experiments, we found that these additional training examples consistently resulted in minor performance improvement for the bar chart class. The inclusion of non-chart images is also crucial since we are tackling a slightly more complex scenario than chart identification, which was the focus in (Savva et al. 2011;

[3] To avoid zero values, we incremented each variable by one before computing its natural logarithm.

Jung et al. 2017). Our model needs to be able to distinguish between posts with images that are not visualisations, and those that indeed contain charts. We appended examples for the non-chart class to the corpus by randomly sampling images from the ILSVRC-2012 dataset (Krizhevsky, Sutskever, and Hinton 2012). For every two chart images, we sampled three random images from ILSVRC-2012—we were keen to create a dataset that is more realistic with respect to the distribution of classes without biasing it heavily in favour of the non-chart class. The resulting corpus, which we refer to as ReVision+, has a total of 6061 images (i.e. 2425 chart images distributed across 11 categories and 3636 non-charts images).

4.2 Collecting Data Visualisations from Twitter

We built a list of 20 Twitter accounts dedicated to data-driven journalism (e.g. GuardianData[4] and nytgraphics[5]). The list was formed in a manual fashion as follows. Many of the major news agencies have Twitter accounts that focus on data visualisations. We browsed the timeline of those accounts, and we followed the suggestions by Twitter (i.e. the "You may also like" section) about accounts with similar content. We included accounts for which we empirically found that $\geq 40\%$ of their shared content included charts.

We then collected the most recent timeline[6] of each one of the 20 accounts using the Twitter API, discarding all tweets without images. When a tweet included multiple images, we stored each unique combination of $e^{(a,m,\mathbf{x})}$ (see Section 3.2) separately. The resulting corpus consists of 34491 tweets. We split this dataset into two parts. The first consists of 3000 messages, whose images we annotated using crowdsourcing. The second was set aside for the initial training of our multi-modal architecture on virality prediction. We refer to them as DataTweet+ and DataTweet respectively.

Building a Realistic Data Visualisation Corpus We ran an experiment on the Figure Eight platform[7] in order to identify the type of charts that are depicted in in the DataTweet+ dataset. In addition to the 3000 randomly selected images tweets, we manually annotated a set of 50 images, which we included as gold standard.

Labelling tasks are designed as so-called Human Intelligence Tasks (or HITs), the unit of work on paid microtask platforms, such as FigureEight. Each HIT consisted of three images, one of whose was always a gold standard example. For each image, the participants were initially asked whether one or more charts were depicted in it, and could choose between a "Yes" and a "No". When no chart was present, they could continue annotating the other images in the HIT. If they answered affirmatively, they were presented with two follow-up questions: first, the participants had to count the number of charts from the image; then they had to classify them. Classification was designed as multiple-choice with checkboxes—crowdworkers were asked to pick all options that applied. We also included an option for "Other" in case the tweet contained a less common chart type. We collected 5 annotations per image. The participants were payed 1¢ for each image they annotated. Participants who failed to retain an accuracy $\geq 80\%$ on the gold standard examples were excluded from the experiment. As a further quality guarantee, we compared for each answer its total number of selected chart types with its total number of depicted charts. In case the first was greater than the second, the annotation was disregarded. Only images with inter-annotator agreement $\geq 60\%$ were included in the resulting DataTweet+ corpus. The Fleiss' Kappa score for the annotations of the images in DataTweet+ was 0.8741. We make the DataTweet+ corpus publicly available at: https://github.com/pvougiou/Pie-Chart-or-Pizza. Table 1 presents the distribution of chart types in the DataTweet+ corpus. For the experiments presented in this paper, we only retained chart images that displayed a single chart type (i.e. 1142 chart image tweets).

5 Experiments

We started by training our ConvNet on ReVision+ (chart and non-chart images; 12 classes, including a non-chart class). We evaluated the trained model on images from ReVision+ and DataTweet+. We then fine-tuned the ConvNet on the training set of DataTweet+ and re-evaluated performance on its validation and test sets.

We used each of the two trained ConvNets from the previous step (without their classification layer) separately in our multi-modal neural architecture to predict the expected number of likes and retweets for a chart post. We trained our systems using the DataTweet corpus, which consists of both chart and non-chart images. Since we sought to predict the virality of data visualisations on Twitter, we focus our evaluation on charts and we use those from the test set of DataTweet+. Besides exploring the contribution of the learned visual features, we investigate how the inclusion of different components (e.g. author-related features) influences the performance in the task.

All images were reshaped to 192×192. During every training phase, we artificially enlarged the number of available images in each corpus by adopting Krizhevsky, Sutskever, and Hinton's data augmentation practices (Krizhevsky, Sutskever, and Hinton 2012). We generated variations for each image by extracting random 160×160 patches and their horizontal reflections and trained our networks on these patches. At test time, we extracted the centre 160×160 patch. The RGB values of each image were centred and normalised by subtracting the mean and dividing with the standard deviation, over each pixel from the training set of ReVision+.

5.1 Training Details

The ConvNet's training objective is to minimise the mean of the negative log-likelihoods of the predictions for a mini-batch of 80 images. The initialisation weights were sampled from a normal distribution with zero mean and 0.01 variance; biases were initialised with zero. Optimisation was

[4]https://twitter.com/GuardianData

[5]https://twitter.com/nytgraphics

[6]https://developer.twitter.com/en/docs/tweets/timelines/overview. Accessed 27 Aug. 2019.

[7]https://www.figure-eight.com

Table 1: Distribution of chart types in DataTweet+. For instance, there are in total 23 images that include area charts, 19 of which contain solely area charts, 3 area charts along with bar charts, and 1 that contains area charts and line graphs. The numbers in brackets represent images that depicted multiple charts of more than two different types. The left table represents categories whose examples included only a single chart type or no chart at all (i.e. `No-Chart`).

Chart Type	
`Table`	39
`Venn`	2
`No-Chart`	1334

Chart Type	`Area`	`Bar`	`Line`	`Map`	`Scatter`	`Pie`
`Area`	19	3	1	–	–	–
`Bar`	3	378	58 ($1^a, 1^b$)	12 (1^a)	–	7 (1^b)
`Line`	1	58 ($1^a, 1^b$)	258	2 (1^a)	3	1 (1^b)
`Map`	–	12 (1^a)	2 (1^a)	382	–	–
`Scatter`	–	–	3	–	31	–
`Pie`	–	7 (1^b)	1 (1^b)	–	–	33

[a]Image including more than two different chart types (i.e. multiple bar charts, line graphs and a map).
[b]Image including more than two different chart types (i.e. a bar chart, a line graph and two pie charts).

performed using Adam (Kingma and Ba 2014) with a learning rate of 10^{-4}. Due to the limited size of the training data, we found that tuning the regularisation parameters properly was crucial in achieving the best performance. We included an l_2 regularisation term of $5 \cdot 10^{-4}$ in the cost function and introduced a dropout value of 0.5 in the first two fully-connected layers. By contrast to VGGNet, we also used batch normalisation before each non-linear activation function and after each convolutional and fully-connected layer (Ioffe and Szegedy 2015). Training stopped at the iteration after which the validation error did not improve.

Fine-tuning the ConvNet. We adapted the ConvNet that has been trained on ReVision+ to the requirements of DataTweet+ by fine-tuning its parameters. The deeper layers of a neural architecture tend to capture corpus-specific features whereas the top ones identify general extraction patterns (Razavian et al. 2014). Consequently, we chose to "freeze" the parameters of the convolutional layers and tune only the fully-connected layers of our architecture (see Figure 2). We used the same training parameters as the ones described in Section 5.1 except the learning rate, for which we used half of its original value.

Multi-modal Neural Architecture. Besides a ConvNet, this consists of modules that process the author-related cues and the text; and predict the retweet and like counts. We used two layers of 256 bidirectional GRUs, and included the $|X| = 6$k more frequent tokens from the texts. We initialised the weights of the modules processing the textual and author information with random uniform distribution between -10^{-3} and 10^{-3}. Optimisation was performed using Adam with a learning rate of 10^{-4} and a batch size of 100. The training was regularised by an l_2 term of 0.02. Batch normalisation and dropout were introduced after each fully-connected layer. We found that increasing the drop rate of the latter to 0.7 helped with some initial overfitting problems that we experienced.

5.2 Chart Identification Evaluation

We evaluated our performance using the accuracy and weighted F_1 metrics on the validation and test set of ReVision+. Except for Venn diagrams and area charts, which were classified with respective F_1 scores of 72% and 71%, all classes were computed with $F_1 \geq 83\%$. Our model performed well in recognising images whose content is irrelevant to charts with binary $F_1 \geq 98\%$. Furthermore, we compared our performance against the reported accuracy scores in (Savva et al. 2011) and (Jung et al. 2017) on ReVision by excluding instances of box plots and non-charts from the validation and test set of ReVision+. Please note that we report our results on the validation, which is the set for which we optimise our performance, and the test set without performing k-fold cross-validation. This enables us to show how the performance of the single, originally trained ConvNet changes across the different tasks, before and after fine-tuning for both chart identification and virality prediction (see Section 5.3). Our system outperformed both. Notably, it was able to achieve greater accuracy scores than Jung et al.'s approach using less than half of its parameters.

Subsequently, we tested our trained system on DataTweet+. We saw a notable performance drop with respect to both the multi-class and binary classification tasks. This is explained by the increased complexity of identifying the chart-related content in images that contain an increased amount of "noisy elements" (e.g. the bar chart and line graph of Table 1); it is also supported by our lower performance on the binary classification task. However, fine-tuning our architecture on this corpus allowed us to substantially increase our performance in terms of both chart and chart-type identification despite the relative—by deep learning standards (Simonyan and Zisserman 2015)—small size of the available datasets. Our performance after fine-tuning also suggests that the features captured by the top, frozen layers of the architecture were useful to the properly-tuned deeper layers. Table 2 summarises the results.

Table 2: Chart identification results on the validation and test sets. ‡ denotes binary experiments in which we seek to recognise chart images from other general-purpose images (`No-Chart`) whose content is irrelevant to charts.

System		Accuracy		F_1	
		Valid.	Test	Valid.	Test
Ours on ReVision+		95.48	93.61	95.49	93.56
Ours on ReVision+ (‡)		99.01	98.55	99.01	98.57
ReVision	(Savva et al. 2011)	80.00[8]	–	–	–
	(Jung et al. 2017)	76.70 (85.0[??])	–	–	–
	Ours	**90.20**	**85.93**	**91.61**	**86.88**
DataTweet+	Ours (‡)	76.16	74.55	75.86	73.55
	Ours + Tuning (‡)	89.90	88.08	89.84	88.08
	Ours	65.45	65.25	71.72	70.21
	Ours + Tuning	**87.07**	**85.86**	**86.56**	**85.57**

5.3 Virality Prediction Evaluation

We seek to identify a chart image from a social media feed, and, subsequently, predict its potential of going viral. The architecture used for classification enables us: (i) to detect images of charts among other general-purpose images on social media, and (ii) to extract from a given chart image high quality features that are used in virality prediction. For evaluating virality prediction, we assume the classification architecture has already identified a shared image as a chart.

We assessed the performance in terms of Root Mean Square Error (RMSE) and Spearman's rank correlation (ρ) between the predicted and the actual numbers of likes and retweets. We computed the expected lower bound scores by using a baseline based on population statistics. This baseline uses the mean and standard deviation (σ) of the like and retweet counts of the DataTweet+'s training set to sample values from a Gaussian distribution for each item in the test set. Furthermore, we couple the original "A" configuration of VGGNet (Simonyan and Zisserman 2015), pre-trained[9] on ILSVRC, with the modules of our multi-modal architecture that process the textual and author features (see Section 3.2). We train this baseline with the same hyper-parameters as the ones presented in the "Multi-modal Neural Architecture" paragraph of Section 5.1. Please note that we adopt the original specification of VGGNet, and we set the fixed-size of the RGB input image to 224×224. We refer to the visual features that are extracted using the original VGGNet as m_{ILSVRC}.

We trained and evaluated different alterations of our architecture in order to determine the separate contribution of each of the three groups of features (text-, author- and chart-related) on this task. We started by training one system (excluding the modules that process the author- and chart-related cues) solely using the text $\mathbf{x}$ of each post. We progressively added more signals and their processing modules. Finally, we compared our predictions using the visual features ($m_{\text{DataTweet+}}$) that were extracted by the fine-tuned ConvNet to the ones with the features from the ConvNet that was trained only on ReVision+ ($m_{\text{ReVision+}}$) and from VGGNet on ILSVRC (m_{ILSVRC}).

[9] https://pytorch.org/docs/stable/torchvision/models.html

Table 3: RMSE (lower is better) in log scale and ρ (higher is better) for like and retweet count predictions on the chart images from the test set of DataTweet+. Rows that start with the + sign refer to systems that process the textual message that accompanies a particular tweet along with one additional cue. For instance, "+ Author (a)" refers to the results from processing both textual and author-related signals. The average performance of the stats baseline along with its σ is reported after sampling 100 times.

System	# of Likes		# of Retweets	
	RMSE	ρ	RMSE	ρ
Stats Baseline	1.545 (.07)	.008 (.064)	1.764 (.07)	.007 (.064)
Text ($\mathbf{x}$)	.955	.371	1.087	.402
+ Author (a)	.899	.479	1.005	.490
+ Chart (m_{ILSVRC})	.951	.421	1.081	.411
+ Chart ($m_{\text{ReVision+}}$)	.917	.450	1.018	.498
+ Chart ($m_{\text{DataTweet+}}$)	.911	.429	1.006	.494
$\mathbf{x}+a+m_{\text{ILSVRC}}$	.861	.527	.974	.528
$\mathbf{x}+a+m_{\text{ReVision+}}$	.862	.507	.969	.536
$\mathbf{x}+a+m_{\text{DataTweet+}}$	**.858**	**.532**	**.962**	**.554**

The findings are summarised in Table 3. We note that considering all available features resulted in the best possible performance. In particular, combining the text with the author cues led to a higher performance gain for the number of likes than with any of the visual features. However, using author- and visual-related features from $m_{\text{ReVision+}}$ and $m_{\text{DataTweet+}}$ boosted almost equally the rank correlation for the number of retweets. In all different combinations, the inclusion of the fine-tuned $m_{\text{DataTweet+}}$ features meant a lower average RMSE compared to the $m_{\text{ReVision+}}$ ones.

Despite the much lower computational complexity of the visual component of the systems equipped with the $m_{\text{DataTweet+}}$ features (around 129M less parameters than VGGNet's "A" configuration), they perform better in both retweet and like prediction than the ones equipped with m_{ILSVRC}. This performance difference is most notable for retweet prediction. This finding highlights the fact that a

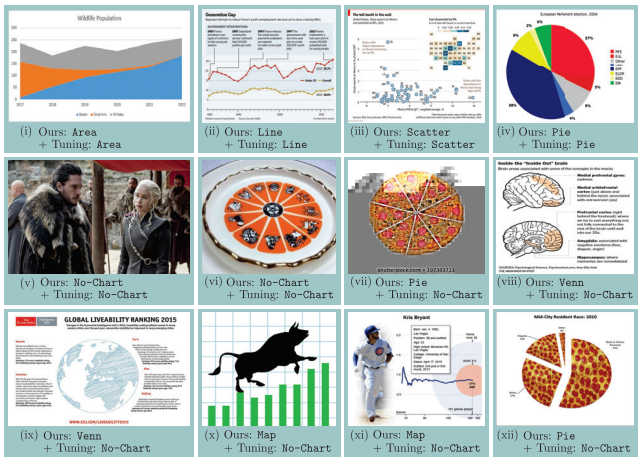


Figure 4: Classification results from testing our originally trained system along with its fine-tuned version (+ Tuning) against different image examples on the Web.

chart is fundamentally different from the general-purpose images included in ILSVRC. The features captured at the latter layers when training on ILSVRC contribute to the recognition of characteristics (e.g. an elephant's tusk) from the 1000 classes of interest. Accurate classification of general-purpose images is performed on the basis of those characteristics. Such characteristics, however, are not dominant in our task. Training and fine-tuning on a chart identification task enables our architecture to implicitly learn to capture the characteristics (e.g. the density of lines and the colour-coding) that are determinant of a particular chart type. We believe that the improved performance of the systems equipped with the $m_{\text{DataTweet+}}$ features indicates that capturing such characteristics can result in more accurate predictions about the virality potential of chart-driven content.

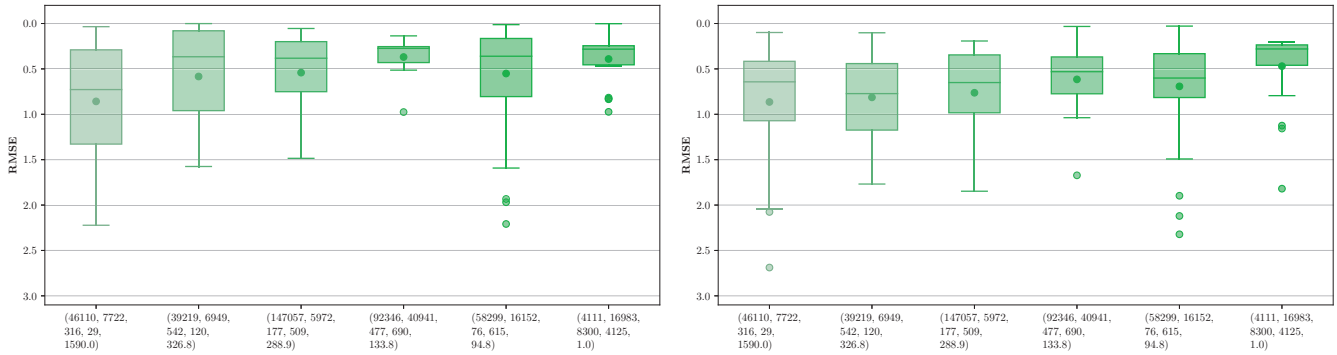
6 Discussion

Our focus in this work was to propose a pipeline with respect to both dataset creation and model design that would enable us to predict the virality potential of chart-driven messages. Our results in Section 5.3 highlight the different levels of importance of the various cues (i.e. related to the text, data visualisation and author) in this prediction task. The implicit features that our architecture learns to capture when it is tuned to the chart identification task result in better performance for virality prediction than a much more computational expensive visual recognition module which has been trained on a collection images larger by two orders of magnitude than ours (Simonyan and Zisserman 2015). We believe that exploring the particular chart characteristics that make a chart go viral is an extremely promising direction for future work. This research direction is in line with recent works that have sought to measure the similarity of different scatter plots either by grouping together charts depicting similar patterns (Abbas et al. 2019) or by leveraging information based on human visual perception in order to learn subjective similarity features (Ma et al. 2020). Their findings along with any additional visual quality metrics, such as blurriness, image resolution and colour-coding, (Antol et al. 2015; Behrisch et al. 2018) could be used to form a list of particular requirements according to which an automatic system would generate data visualisations. Our virality prediction architecture could be used to estimate the expected number of retweet and like counts of these visualisations, for the same, artificially selected, author and accompanying text message. Based on these predictions, we would be able to better understand chart characteristics that result in higher retweet or like counts. We see our DataTweet+ dataset as an important step towards this direction—both with respect to the qualitative (e.g. general format of data visualisations shared on social media) and quantitative (e.g. training data, distance between the automatically generated charts and the empirical ones from our corpus) analysis required for the design of the above generator.

We opted to evaluate the two parts of our proposed pipeline (i.e. chart identification and virality prediction) separately to better explore their individual strengths and limitations. The tight relation between these two tasks becomes more apparent in Table 3, where the fine-tuned features of the model trained on the charts from DataTweet+ enable us to achieve the best performance for virality prediction. Through our training strategy (i.e. training on ReVision+ and fine-tuning on DataTweet+), our ConvNet learns to capture the visual characteristics that not only differentiate a data visualisation from a general-purpose image but also classify it according to its depicted chart type. Our results in Section 5.3 highlight that these learned features are immediately applicable to virality prediction since the ConvNets that are trained on the more relevant datasets (i.e. ReVision+ and DataTweet+) achieve better performance than the original VGGNet—a much more computational expensive system which has been trained on a collection images larger by two orders of magnitude than ours.

We tested both chart identification ConvNets against a representative set of images from the web to gain a better understanding of their performance differences. The findings are summarised in Figure 4. We note that examples of images that can be clearly categorised to one of the 11 chart types or to the non-chart class are correctly classified by both systems (i.e. Figures 4i, 4ii, 4iv, 4v and 4vi). In addition, fine-tuning the system makes it more efficient at categorising more borderline cases of images, such the ones presented in Figure 4vii, 4viii, 4ix and 4x, that resemble a chart without being one. This is in line with the performance on the binary task reported in Table 2, where the fine-tuned variant achieved an improvement of at least 14% in F_1 score in recognising images whose content is irrelevant to charts, compared to the initial system.

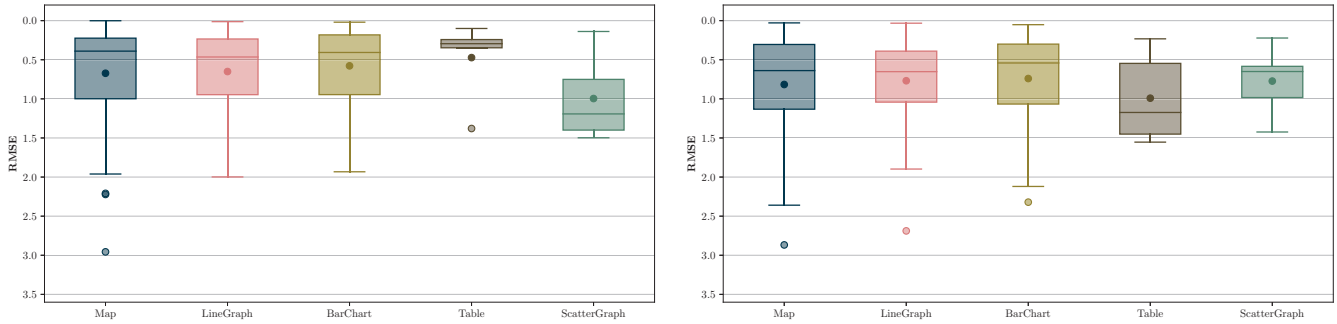
While the accuracy with which charts were identified has improved considerably after fine-tuning our ConvNet on DataTweet+, there were still images that posed challenges to the task. These tend to fall into one of these three categories: (i) dashboard visualisations that consist of more than a single chart type (e.g. Figure 4iii); (ii) charts that are embedded in images with complex text and graph-



(a) # of Likes

(b) # of Retweets

Figure 5: Average RMSE in log scale of our fine-tuned system ($\mathbf{x} + a + m_{\text{DataTweet+}}$) across different authors. Each author is presented using their corresponding features: $(a^{(f)}, a^{(s)}, a^{(l)}, a^{(b)}, a^{(r)})$.



(a) # of Likes

(b) # of Retweets

Figure 6: Average RMSE in log scale of our fine-tuned system ($\mathbf{x} + a + m_{\text{DataTweet+}}$) across different chart types. Only chart types with at least 5 examples in the test set of DataTweet+ have been included in this evaluation.

ics layouts (e.g. Figure 4xi); and (iii) charts with a high share of "chartjunk", which use textures, illustrations and background imagery rather than standard visual representations of data (Tufte 1986) (e.g. Figure 4xii). While the crowdsourcing pipeline we implemented could handle images with single or multiple charts and chart types with high confidence, our experiments focused on single chart-type examples. The ConvNet models presented earlier could be naturally extended to predict both the number of charts and their type to cover. Furthermore, we could expand the DataTweet+ dataset to include examples of visualisations from the other two categories. This would help us tune the model better to address more borderline scenarios.

Almost 99% of the DataTweet+ corpus was made of images that were at least six months old. As many as 95% of the images were older than a year. Hence, we expected that the retweet and like rate of each message would have been almost zero at data collection time. Since our goal was to predict the aggregated exposure that a chart message would get, we built upon previous work on predicting image virality (Can, Oktay, and Manmatha 2013; Khosla, Das Sarma, and Hamid 2014), and opted not to model time in our architecture. Exploring retweets and likes as a function of time would be an interesting direction of future work, and we believe that our approach could serve as a starting point. A trivial method for parameterising our model over the "age" of a chart post would be to introduce it as an additional input, concatenated with the author features (see Section 3.2).

In addition to the virality experiments presented in Section 5.3, we compared the predictions of our best performing system (i.e. $\mathbf{x} + a + m_{\text{DataTweet+}}$ in Table 3) to the actual numbers of retweets and likes across authors who posted at least 10 chart images (see Figure 5). Each author in Figure 5 is presented using their corresponding features: $(a^{(f)}, a^{(s)}, a^{(l)}, a^{(b)}, a^{(r)})$. We note that the RMSE was lower the closer the ratio between number of followers and friends, $a^{(r)}$, was to one. The variability of the error values was also less for the authors with a lower $a^{(r)}$. This means that the predictions for the expected number of retweets and likes of a particular chart post deviated less from their actual values when $a^{(r)}$ was almost 1.

Figure 6 shows the RMSE of the predicted number of

retweets and likes across chart types with at least 5 examples in the test set of DataTweet+. We note that there were only minor deviations in the mean RMSE for predicting the two metrics across chart types. This means that our end-system was capable of making predictions for chart types, which based on the mean values of their corresponding messages, have both high (e.g. scatter graphs and maps) and low (e.g tables) expected number of retweets and likes.

7 Conclusion

To the best of our knowledge, this work constitutes the first attempt to estimate the virality of data visualisations on social media. We proposed an end-to-end learnable approach that identifies images of charts as they are posted on social media, and predicts their virality potential. We believe our work is a first step towards addressing the propagation of fake news through charts on social media; a chart which is expected to gain significant exposure could be subjected to further inspections regarding the accuracy of its content.

In our experiments, we did not investigate the identification of charts in images that display more than one chart types. Nonetheless, we found that examples of such images tend to be relatively common among data journalists' posts (as shown in Table 1). A natural extension of this work is the implementation of an architecture capable of predicting both the number of depicted charts and their type. As noted briefly in the discussion, the ConvNets trained on ReVision+ and DataTweet+ could be easily extended to predict the number of charts in an image as well.

Our system and findings could be used in different scenarios—from generating automatic text captions and recommending chart improvements in data visualisation tools to informing marketing strategies for brands that use data visuals to gauge customer engagement. In addition, our approach, including both the neural architecture and the method to create labelled data, could form the basis for the development of visual question answering solutions tailored to data visualisations, with applications in fact checking and misinformation online.

Acknowledgements

We thank the reviewers for their thorough and insightful feedback. This research is partially supported by the Data Stories project, funded by EPSRC research grant No. EP/P025676/1. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research.

References

Abbas, M. M.; Aupetit, M.; Sedlmair, M.; and Bensmail, H. 2019. Clustme: A visual quality measure for ranking monochrome scatterplots based on cluster patterns. *Computer Graphics Forum* 38(3):225–236.

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. VQA: Visual question answering. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 2425–2433.

Behrisch, M.; Blumenschein, M.; Kim, N. W.; Shao, L.; El-Assady, M.; Fuchs, J.; Seebacher, D.; Diehl, A.; Brandes, U.; Pfister, H.; Schreck, T.; Weiskopf, D.; and Keim, D. A. 2018. Quality metrics for information visualization. *Computer Graphics Forum* 37(3):625–662.

Can, E. F.; Oktay, H.; and Manmatha, R. 2013. Predicting retweet count using visual cues. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, 1481–1484. New York, NY, USA: ACM.

Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734. Doha, Qatar: Association for Computational Linguistics.

Cogswell, M.; Ahmed, F.; Girshick, R. B.; Zitnick, L.; and Batra, D. 2015. Reducing overfitting in deep networks by decorrelating representations. *CoRR* abs/1511.06068.

Deza, A., and Parikh, D. 2015. Understanding image virality. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gray, J.; Chambers, L.; and Bounegru, L. 2012. *The Data Journalism Handbook: How Journalists Can Use Data to Improve the News*. O'Reilly Media, Inc., 1st edition.

Guerini, M.; Staiano, J.; and Albanese, D. 2013. Exploring image virality in google plus. In *Proceedings of the 2013 International Conference on Social Computing*, SOCIALCOM '13, 671–678. Washington, DC, USA: IEEE Computer Society.

Huang, W., and Tan, C. L. 2007. A system for understanding imaged infographics and its applications. In *Proceedings of the 2007 ACM Symposium on Document Engineering*, DocEng '07, 9–18. New York, NY, USA: ACM.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F., and Blei, D., eds., *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, 448–456. Lille, France: PMLR.

Jung, D.; Kim, W.; Song, H.; Hwang, J.-i.; Lee, B.; Kim, B.; and Seo, J. 2017. Chartsense: Interactive data extraction from chart images. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, 6706–6717. New York, NY, USA: ACM.

Khosla, A.; Das Sarma, A.; and Hamid, R. 2014. What makes an image popular? In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, 867–876. New York, NY, USA: ACM.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet classification with deep convolutional neural networks. In Pereira, F.; Burges, C. J. C.; Bottou, L.; and Wein-

berger, K. Q., eds., *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc. 1097–1105.

Ma, Y.; Tung, A. K. H.; Wang, W.; Gao, X.; Pan, Z.; and Chen, W. 2020. Scatternet: A deep subjective similarity model for visual analysis of scatterplots. *IEEE Transactions on Visualization and Computer Graphics* 26(3):1562–1576.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. Doha, Qatar: Association for Computational Linguistics.

Prasad, V. S. N.; Siddiquie, B.; Golbeck, J.; and Davis, L. S. 2007. Classifying computer generated charts. In *2007 International Workshop on Content-Based Multimedia Indexing*, 85–92.

Razavian, A. S.; Azizpour, H.; Sullivan, J.; and Carlsson, S. 2014. CNN features off-the-shelf: An astounding baseline for recognition. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, CVPRW '14, 512–519. Washington, DC, USA: IEEE Computer Society.

Savva, M.; Kong, N.; Chhajta, A.; Fei-Fei, L.; Agrawala, M.; and Heer, J. 2011. ReVision: Automated classification, analysis and redesign of chart images. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, 393–402. New York, NY, USA: ACM.

Shao, M., and Futrelle, R. P. 2006. Recognition and classification of figures in pdf documents. In Liu, W., and Lladós, J., eds., *Graphics Recognition. Ten Years Review and Future Perspectives*, 231–242. Berlin, Heidelberg: Springer Berlin Heidelberg.

Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tufte, E. R. 1986. *The Visual Display of Quantitative Information*. Cheshire, CT, USA: Graphics Press.

Zhao, Z.; Meng, L.; Xiao, J.; Yang, M.; Wu, F.; Cai, D.; He, X.; and Zhuang, Y. 2018. Attentional image retweet modeling via multi-faceted ranking network learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 3184–3190. International Joint Conferences on Artificial Intelligence Organization.