

Insights on the V3C2 Dataset

Luca Rossetto
University of Zurich
rossetto@ifi.uzh.ch

Abraham Bernstein
University of Zurich
bernstein@ifi.uzh.ch

Klaus Schoeffmann
University of Klagenfurt
ks@itec.aau.at

ABSTRACT

For research results to be comparable, it is important to have common datasets for experimentation and evaluation. The size of such datasets, however, can be an obstacle to their use. The Vimeo Creative Commons Collection (V3C) is a video dataset designed to be representative of video content found on the web, containing roughly 3 800 hours of video in total, split into three shards.

In this paper, we present insights on the second of these shards (V3C2) and discuss their implications for research areas, such as video retrieval, for which the dataset might be particularly useful. We also provide all the extracted data in order to simplify the use of the dataset.

KEYWORDS

Dataset, Video Collection, Video Analytics, Content Statistics, Video Browser Showdown, TRECVID

1 INTRODUCTION

Common datasets form the basis for comparable research by providing a standardized foundation to evaluate methods on and compare their relative behavior. With the ever increasing amount of data produced, especially in the multimedia domain, such datasets also become increasingly larger, especially when they not only cover a highly specialized content area but try to be representative of a larger whole. The *Vimeo Creative Commons Collection (V3C)* [9] is a video dataset designed to be representative of video content found on the web. The full dataset is split into three shards with a total combined duration of 1 000, 1 300 and 1 500 hours respectively. The first shard (V3C1) has been used since 2019 for several evaluation campaigns, including the Video Browser Showdown (VBS) [8] and the TRECVID [1] Ad-hoc Video Search (AVS) Task. While excerpts from the second shard (V3C2) have already been used in recent TRECVID Video-to-Text (VTT) Tasks, the full content of V3C2 is scheduled to enter wider use in 2021. In particular, V3C2 is going to be used for TRECVID AVS 2022 as well as for VBS 2022, and subsequent upcoming iterations of both competitions. In preparation, analogously to prior work (such as [2] or [7]), we present an overview of the properties and contents of this dataset and provide additional resources to facilitate its use. To that end, Section 2 provides an overview of the more technical aspects of the dataset while Section 3 discusses some properties of its content. Section 4 then concludes the insights and discusses their consequences for the usefulness of the dataset in different research areas. In addition to the presented overview, we make the data generated during this analysis available in order to provide an easier start when using the dataset.¹

¹<https://github.com/lucaro/V3C2Analysis>

2 V3C2 OVERVIEW

In this section, we analyse several statistics of technical properties of the V3C2 dataset and compare their distribution to those of V3C1. Similarly to the analysis presented in [2], we examine the distribution of several properties of V3C2 and compare them with their respective counterparts from V3C1. The dataset is composed of videos collected from the video sharing platform Vimeo² which were released under a creative commons license and have a duration between three and 60 minutes. Figure 1 shows the cumulative distribution of the overall video duration for both V3C1 and V3C2. Both distributions are closely aligned (with a Pearson correlation of 0.939 before cumulative aggregation) and shorter videos clearly dominate both shards, with roughly 80% of videos having a duration of at most 10 minutes.

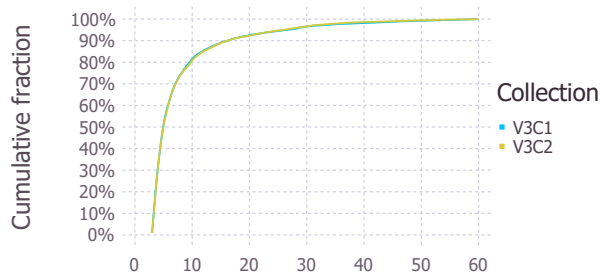


Figure 1: Distribution of video duration in minutes

Figure 2 shows the distribution of the duration across the 1 425 451 segments of V3C2, overlaid onto the same distribution from V3C1. It can be seen that both lines have the same overall shape (with a correlation of 0.999) and exhibit only minor differences. Each video in the second shard of the dataset has roughly 146 segments on average, with a minimum of 4 and a maximum of 5 814.



Figure 2: Distribution of segment duration in seconds

Figure 3 compares the age distribution of the videos, as indicated by their date at which they have been uploaded to Vimeo,

²<https://vimeo.com>

Table 1: Occurrence of file formats

Video Container	Count (Fraction)	
	V3C1	V3C2
MP4	7358 (98.43%)	9642 (98.79%)
MOV	102 (1.36%)	107 (1.1%)
M4V	11 (0.15%)	8 (0.08%)
AVI	1 (0.01%)	3 (0.03%)
MPE	3 (0.04%)	0 (0%)

between the two shards of the dataset. The figure shows the upload time-stamps aggregated into two-week intervals. Both curves have a similar overall shape, which is however rather noisy (Pearson correlation coefficient: 0.233).



Figure 3: Distribution of video upload over time

Figure 4 shows a histogram of the number of pixels per frame, with common resolutions indicated on the horizontal axis. It can be seen that both shards have rather similar distributions (Pearson correlation coefficient: 0.488), with the most commonly occurring video resolution being 1280×720 pixels, followed by 1920×1080.

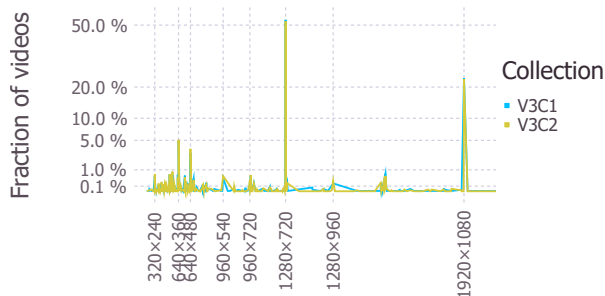


Figure 4: Distribution of video resolution

Since Vimeo preserves at least some of the original uploaded video files and the videos have not been re-encoded when the dataset was compiled, it is not homogeneous with respect to codecs and containers used for encoding. Table 1 provides an overview of the video containers as identified by their file endings. For both V3C1 and V3C2, the vast majority of videos use MP4 as a container, with the second most popular being MOV, followed by M4V.

3 V3C2 CONTENT

A high-level insight into the content of the videos contained within the dataset can be obtained by looking at the categories to which the videos have been assigned by their creators during the upload to Vimeo. Figure 5 shows the top-10 assigned category labels for both V3C1 and V3C2. It can be seen that not only are the top-10 categories the same for both datasets, they are also in the same order. Considering all 71 categories that are present in both shards, the Spearman rank correlation between them is 0.987.

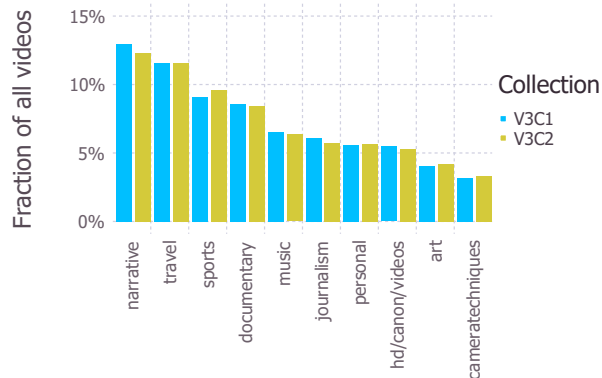


Figure 5: Top-10 Vimeo categories

3.1 Low-level Visual

Table 2 shows the dominant colors of the video keyframes, which were made available as part of the dataset. When comparing the distribution of dominant colors across the two shards of the dataset, it can be seen that they are almost identical, despite this not being given any consideration during the creation of the dataset [9]. For both datasets, almost 23% of keyframes do not have a dominant color, meaning that no color covers at least half the pixels in the frame. The next largest categories are predominantly gray frames, having a majority of pixels with a saturation of between 2% and 20% or a value of 30% or below in the HSV color space, followed by black and white frames – which means that the majority of pixels have a color saturation of below 2%. The most common dominant hues (h) with a saturation above 20% are orange ($0.07 \leq h < 0.14$), red ($h < 0.07 \vee h \geq 0.92$) and blue ($0.65 \leq h < 0.73$), followed by green ($0.17 \leq h < 0.44$), cyan ($0.44 \leq h < 0.56$) and magenta ($0.76 \leq h < 0.92$).

3.2 High-level Visual

In order to get a sense of the semantic variability of the dataset, we use the last layer of an InceptionResNet [11] which was pre-trained on ImageNet [4] to encode the visual semantic information of the videos. The feature representation is generated for each frame and mean-pooled by segment for both V3C1 and V3C2. For the resulting 2 508 108 feature vectors, the pair-wise L1 distance is computed. Figure 6 shows a histogram of the resulting distances for both shards of the dataset individually as well as in combination.

It can be seen that the three histograms are all rather well aligned (with a μ of 45.918, 46.227, and 46.096 a σ of 6.493, 6.490, and 6.491,

Table 2: Dominant Colors of Shot keyframes

Dominant Color	Key-frame count (ratio)	
	V3C1	V3C2
Blue	32 058 (2.96%)	40 024 (2.81%)
Cyan	8 109 (0.75%)	11 032 (0.77%)
Green	10 912 (1.01%)	13 665 (0.96%)
Magenta	1 899 (0.18%)	3 061 (0.21%)
Orange	33 188 (3.07%)	48 683 (3.42%)
Red	30 569 (2.82%)	40 127 (2.82%)
Violet	108 (0.01%)	152 (0.01%)
Yellow	1 424 (0.13%)	1 812 (0.13%)
Black/White	94 769 (8.75%)	114 199 (8.01%)
Gray	624 835 (57.71)	827 662 (58.06%)
None	244 789 (22.61%)	325 040 (22.80%)

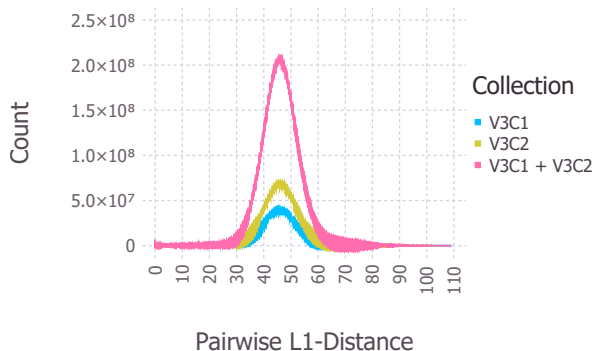


Figure 6: Distribution of pairwise distances of semantic features extracted using an InceptionResNet.

respectively), indicating that the semantic variability (as measured by the used neural network) within each collection has a similar distribution than the combination of both shards.

3.3 Detected Objects

In order to get an overview of the semantic content in the dataset, we also performed object detection with the YOLOv4 [3] detector (using Darknet-53). For this, we simply used the publicly available model that was pre-trained on 80 classes of MS COCO [6]. The process detected a total of 5 009 059 object instances, or roughly 3.5 instances per keyframe on average. This average value is above the median of 2 detected objects per keyframe, since for 334 221 or about 23.4% of all keyframes, none of the 80 object classes were detected. Table 3 shows the top-10 most commonly detected objects and their occurrence.

3.4 Faces

We further applied face detection to every keyframe by using Facenet [10]. An overview of the distribution of detected faces is shown in Table 4. While the number of individual faces detected in a keyframe ranges from 0 to 289, the majority of keyframes has no detectable face. For the frames with detections, the majority

Table 3: Top-10 detected object instances

Object	Count	Object	Count
Person	3 102 682	Bottle	79 078
Car	266 083	Cup	69 745
Chair	164 530	Bicycle	61 765
Book	104 849	Bird	61 177
Tie	95 933	Boat	51 442

has only one face. The total number of detected face instances is 1 352 749, or roughly 43.6% of the number of detected people as shown in the previous section.³

Table 4: Detected faces per keyframe

Number of Faces	Keyframe count (ratio)
0	848 099 (59.5%)
1	329 808 (23.1%)
2	114 864 (8.1%)
3	49 582 (3.5%)
4	25 842 (1.8%)
≥ 5	57 257 (4%)

3.5 Scene Text

To analyze the occurrence of text within the videos, we applied a scene text detection and transcription pipeline on all the pre-extracted keyframes of V3C2. The pipeline uses easyOCR⁴ in order to detect and extract visible text written using the Latin, Persian, and Thai alphabet as well as Simplified Chinese, Devanagari, Japanese, or Hangul characters.

For the 9 760 videos in V3C2, the pipeline identified text in at least one key-frame in 9 467 of them. Figure 7 shows the distribution of the fraction of keyframes per video. It can be seen that while many videos only have comparatively few segments for which text was detected, the distribution is rather long-tailed, indicating a substantial subset of videos for which text occurs throughout a large fraction of the video. However, of the 1 425 451 segments, only 425 919, or roughly 29.88% contain any detected text. The mean number of keyframes with text per video is roughly 43.64, the median is 21.

Figure 8 shows a distribution of the number of individual detected text items for all frames with any detected text. The used OCR pipeline groups detected text into items if all letters are sufficiently close to each other. It can be seen that here, the distribution is even more skewed, with the vast majority of the frames only having very few detected items, while only a small number of frames have many text items. For all frames for which any text was detected, roughly 57.43% have at most 5 detected text items and 77.89% have 10 or fewer.

³This discrepancy is mostly explained by the fact that the object detector is also capable of detecting people in scenes where their faces are not visible.

⁴<https://github.com/JaidedAI/EasyOCR>

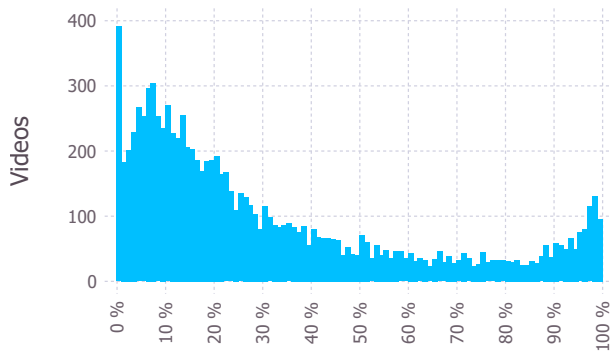


Figure 7: Distribution of fraction of keyframes with text per video

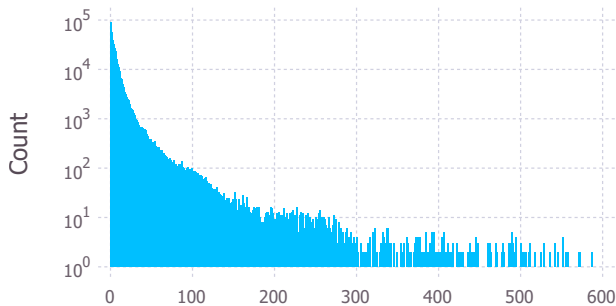


Figure 8: Distribution of detected text items per video

3.6 Speech

In order to analyze the spoken components of the audio found in the dataset, we apply Mozilla’s implementation⁵ of DeepSpeech [5] in version 0.9.3 to the entire second shard of the dataset. The ASR pipeline was prefaced by a voice activity detector⁶ in order to reduce the amount of incorrectly transcribed non-voice audio. This process produces a list of triples for every video, which contain the start and end timestamp of a segment with detected voice activity as well as the transcribed utterances within this interval. The transcript is always performed under the assumption, that all speech is English.⁷ Since these temporal segments are generated only based on voice activity, they are not aligned with the master shot references provided with the dataset. Segments for which voice activity was detected but no transcript could be generated were removed. The process generated at least one segment with a transcript for 9 613 out of the 9 760 videos of V3C2. For these videos, voice activity was detected during on average 90.3% of the video, resulting in 20.225 individual segments per video on average. Figure 9 shows the distribution of the detected number of spoken words per minute for all speech intervals for all videos for which any speech was transcribed.

⁵<https://github.com/mozilla/DeepSpeech>

⁶analogously to the pipeline given in https://github.com/mozilla/DeepSpeech-examples/tree/r0.9/vad_transcriber

⁷This is certainly not true, but in the absence of a reliable method for language identification, it serves as a first-order approximation.

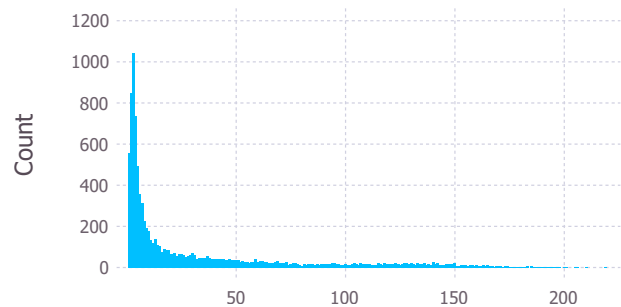


Figure 9: Distribution of detected spoken words per minute

Both the high fraction of detected speech activity as well as the skewed distribution of words per minute indicate, that the used voice activity detector miss-labeled a sizable fraction of non-speech activity as speech. The figure also indicates that there are only few videos with more or less continuous dialog, while the majority of videos appear to only have occasional spoken words.

4 CONCLUSION

The analyses presented above show that there is a large amount of variation to be found within the dataset — both on a technical as well as a semantic level and across the modalities. Especially relevant in the area of content-based retrieval, for which the dataset is primarily used, is that none of the methods which generate text can be relied on to index the entire dataset. While text-based queries targeting scene-text or transcripts have been previously shown to be a very effective way of retrieving video sequences (especially in a known-item search setting) neither text generation method covers the dataset to a sufficient degree to be able to rely on them exclusively.

The large amount of variation also indicates that the dataset could, potentially using some additional annotations, be used for research areas other than retrieval.

With the presented analysis of several aspects of V3C2, we offer insights into its content in order to help researchers estimate if the dataset could be useful for their work. All the generated data on which the presented analysis is based is made available, in order to simplify future use of the dataset.

A REPOSITORY OVERVIEW

All the data on which the presented analyses are based is available via <https://github.com/lucaro/V3C2Analysis>. The larger resources, including the shot-level features used for the analysis presented in Section 3.2 as well as the detected object instances discussed in Section 3.3 are available via <https://files.ifi.uzh.ch/ddis/v3c/>.

Figure 10 shows an overview of the repository’s structure and the data found within. The repository also contains some of the code which was used extract data from the videos and keyframes.

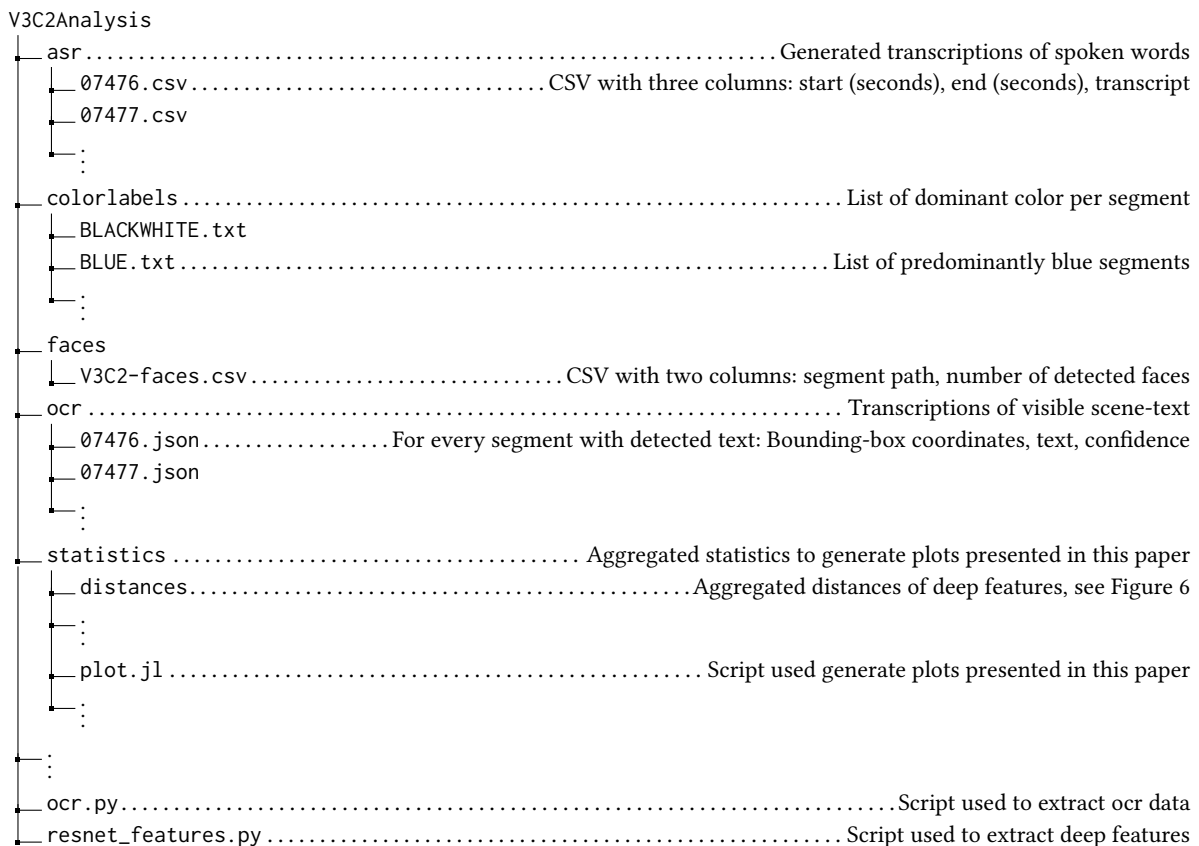


Figure 10: Structure of the repository containing the supplementary material

REFERENCES

- [1] George Awad, Asad A. Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, Andrew Delgado, Jesse Zhang, Eliot Godard, Lukas Diduch, Jeffrey Liu, Alan F. Smeaton, Yvette Graham, Gareth J. F. Jones, Wessel Kraaij, and Georges Quénot. 2020. TRECVID 2020: comprehensive campaign for evaluating video retrieval tasks across multiple application domains. In *Proceedings of TRECVID 2020*. NIST, USA.
- [2] Fabian Berns, Luca Rossetto, Klaus Schoeffmann, Christian Beecks, and George Awad. 2019. V3C1 Dataset: An Evaluation of Content Characteristics. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval, ICMR 2019, Ottawa, ON, Canada, June 10-13, 2019*. ACM, 334–338. <https://doi.org/10.1145/3323873.3325051>
- [3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* (2020).
- [4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.
- [5] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Sathesh, Shubho Sengupta, Adam Coates, et al. 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567* (2014).
- [6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.
- [7] Paul Over, George Awad, Alan F Smeaton, Colum Foley, and James Lanagan. 2009. Creating a web-scale video collection for research. In *Proceedings of the 1st workshop on Web-scale multimedia corpus*. 25–32.
- [8] Luca Rossetto, Ralph Gasser, Jakub Lokoc, Werner Bailer, Klaus Schoeffmann, Bernd Muenzer, Tomas Soucek, Phuong Anh Nguyen, Paolo Bolettieri, Andreas Leibetseder, et al. 2020. Interactive video retrieval in the age of deep learning-detailed evaluation of vbs 2019. *IEEE Transactions on Multimedia* 23 (2020), 243–256. <https://doi.org/10.1109/TMM.2020.2980944>
- [9] Luca Rossetto, Heiko Schuldt, George Awad, and Asad A. Butt. 2019. V3C - A Research Video Collection. In *MultiMedia Modeling - 25th International Conference, MMM 2019, Thessaloniki, Greece, January 8-11, 2019, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 11295)*. Springer, 349–360. https://doi.org/10.1007/978-3-030-05710-7_29
- [10] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 815–823.
- [11] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. 2016. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261* (2016).