Detection of cats in images

ANNA MARIA WALACH (s121540)
Technical University of Denmark
02238 Biometric Systems

The LaTeX `acmtrans` document style formats articles in the style of the ACM transactions. Users who have prepared their document with LaTeX can, with very little effort, produce camera-ready copy for these journals.

Categories and Subject Descriptors: D.2.7 [**Software Engineering**]: Distribution and Maintenance—*documentation*; H.4.0 [**Information Systems Applications**]: General

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: cat, detection, images

1. INTRODUCTION

There are more and more projects that uses live streaming and social media power to help and safe feral cats. Some of them are meant to control the population in the area [Society 2015], other focus on raising awareness about feral cats, spaying and neutering importance [Kittens 2015] or just to increase changes of finding a new home for homeless cats and kittens [Room 2015].

People are also more interested in health and safety of their own pets. Lower prices of video technologies and smarthone popularity helped to create various apps for controlling your pet lifestyle. [Zillians 2015] created an automatic feeder with cat identification system and connected it to the smartphone app. It allows owner to supervise the amount of dry food and water his cats are consuming everyday, to receive alarms in case of abnormalities and even spy on them while eating.

Fig. 1. CatFi product used by a cat.

The PiP [Company 2015] app is designated to help people find their lost cats and dogs. According to American Human Society, almost 3.5 million pets are lost each year and about 80% of dogs and 98% of cats are never re-united with their families. The authors of the smarthone app allows the owners to made profiles for their pets (including pictures) and in case the pet is missing, you launch the "alarm" on that pet, together with last seen location. If anyone finds a homeless animal, he may send the photo of it to the PiP app, which automatically try to match one of the lost pets with found pet.

Increasing number of pets' photographies also encouraged camera producers to implement appropriate support for it in their devices. Right now, Fujifilm company

introduced Auto Dog / Cat Detection function [FujiFilm 2015] in some of their cameras that allows to automatically detect pet's face on the image and put auto-focus on it.

1.1 Problem description

Almost all of the applications and initiatives described in previous section already use or would benefit from the cats detection and/or identification. In this research I would like to focus on topic of cat detection in images. The main interest is on the use case of detecting cats presence in shots from live cameras and streams. Such images are characterized by diversified, often bad lighting settings, with possible presence of another animals or human beings. I will perform a comparison of existing solutions and propose own framework that deals with this problem.

2. THE STATE OF THE ART

Reliable comparison of existing software for detecting cats in pictures is hard to performed. It is caused by a few factors:

—**box algorithms** - many algorithms, that claim to have high accuracy, are part of the hardware product [Zillians 2015] or closed-source app [Company 2015]. This way it is impossible to make a theoretical comparison of the algorithms and usually significantly hampers possibility of performing experiments with test data, as the products' interfaces are not adjusted to bulk operations.

—**high diversity of use cases** - although there are a lot of studies focusing on cat detection, usually the classification or testing environmental does not check robustness of the algorithm in typical real-time scenarios. The classification is e.g. made as cat vs still life [Fleuret and Geman 2007; Akihiko Yamada et al. 2011] or as cat/dog breed detection [Omkar M. Parkhi et al. 2012].

—**different test sets** - even for algorithms with similar use cases, e.g. cat vs still life, different test set are in use to estimate the accuracy of algorithm, which makes it significantly harder to perform direct comparisons. Usually articles use their own data sets [Omkar M. Parkhi et al. 2012; Fleuret and Geman 2007; Akihiko Yamada et al. 2011] and only some of them compare to standard data bases, like Pascal 2007 cats database [Omkar M. Parkhi et al. 2012].

2.1 Comparison of cat vs still life algorithms

3. METHODOLOGY

3.1 Overfull hbox - Stretching/filling one horizontal line

To solve a line break due to "Overfull `\hbox`", here is a plain TEX solution; here `\hsize` is the default setting of acmtrans.sty:

```
\hbox to \hsize{line sentence to be stretched}
```

This can be used in a list environment as well but `\hsize` declared to a reduce dimension:

```
\hbox{\vbox{\hsize = less than the default setting
\hbox to \hsize{line sentence to be stretched}}}
```

```
type date =
  record day: 1..31;
         month: 1..12;
         year: integer
  end
var mybirth, today : date;
var myage : integer;
```

Fig. 2. An example of a program centered in a figure

3.2 Programs

Good formatting of programs requires a knowledge of their semantics, and is beyond the scope of a document production system. While "pretty printers" are useful for handling the many pages of a real program, the short examples that are published in articles should be formatted by hand to improve their clarity. The LaTeX `tabbing` environment makes the formatting of programs relatively easy, especially if the user defines commands for his particular language constructs. One may also use the `verbatim` environment.

The ACM transactions style requires that programs be formatted with different size fonts, depending upon whether they appear in the text or in a figure, but that is handled by the figure macro which automatically sets the correct font size. Moreover, programs in running text should be indented two ems on each side (as provided by the `quote` environment), and programs in regular figures should be centered. (Programs in "narrow figures" (q.v.) are left or right justified automatically).

Here is an example of a program:

```
type date =
  record day: 1..31;
         month: 1..12;
         year: integer
  end
var mybirth, today : date;
var myage : integer;
```

Figure 2 shows how the same program looks in a figure.

In addition to formatting programs, the `tabbing` environment may be used for similar displayed material such as BNF syntax specifications and rewrite rules.

4. FIGURES AND TABLES

4.1 Figures

The ordinary LaTeX `figure` environment works as usual. Figure 3, which is Figure 6 of Nielson [1985], a bogus reference, was produced in this way. Note that figures should never appear in the text or at the bottom of a page. (If you use the figure placement optional argument, use only `t` or `p` or both; do not use `h` or `b`).

Some figures (and tables) have no caption except for the figure number. For such figures (and tables), one uses a `\nocaption` command, which has no argument, instead of the `\caption` command.

	$\perp$	F	T
$\perp$	$\perp$	$\perp$	T
F	$\perp$	F	T
T	$\perp$	T	T

Fig. 3. The truth table for the parallel-or.

Table I. The truth table for the parallel-or.

	$\perp$	F	T
$\perp$	$\perp$	$\perp$	T
F	$\perp$	F	T
T	$\perp$	T	T

In addition to this method of formatting figures, the ACM transactions also uses figures with side captions, as in Figure **??**. Such a figure is produced with the `narrowfig` environment. This environment has a single mandatory argument, which is the width of the figure. Note that if the figure is generated by `tabbing` or `tabular`, one can safely overestimate the size. It works just like the ordinary `figure` environment, except it must contain only one `\caption` or `\nocaption` command, which must come after the figure itself.

The `narrowfig` environment should obviously not be used unless the figure is narrow enough to leave a reasonable amount of space beside it for the caption. The ACM seems to have no consistent policy for choosing which style of figure to employ.

4.2 Tables

The ordinary LaTeX `table` environment can be used, but it requires the user to add formatting commands to match the ACM transactions style. This formatting is performed automatically if the `acmtable` environment is used instead, producing the result shown in Table I, which shows the same table displayed in Figure 3.

4.3 Bibliography

(1) [Cytron et al. 1991] → `\cite{cytron-et-al-toplas91}`

(2) Briggs et al. [1994] → `\citeN{briggs-cooper-torczon-toplas94}` or

The list will be updated as we find unique cases.

REFERENCES

Akihiko Yamada, Kazuhiro Kojima, J. K., Okamoto, M., and Murata, H. 2011. Directional edge-based dog and cat face detection method for digital camera. *IEEE International Conference on Consumer Electronics*.

Company, T. P. R. 2013 (accessed June 22, 2015). *PiP - reuniting lost pets with their families.* `http://www.petrecognition.com/`.

Fleuret, F. and Geman, D. 2007. Stationary features and cat detection. *Idiap Research Report*.

FujiFilm. 2009 (accessed June 22, 2015). *Auto Dogs & Cats Detection.* `http://www.fujifilm.com/products/digital_cameras/pet/`.

Kittens, T. 2015 (accessed June 22, 2015). *Rescue Kitten Reality TV.* `http://www.tinykittens.com/`.

Nielson, F. 1985. Program transformations in a denotational setting. *ACM Trans. Program. Lang. Syst. 7,* 3 (July), 359–379.

Omkar M. Parkhi, A. V., Zisserman, A., and Jawahar, C. V. 2012. Cats and dogs.

Room, T. C. 2015 (accessed June 22, 2015). *Foster Kitten Cam*. `https://livestream.com/FosterKittenCam`.

Society, L. A. P. 2015 (accessed June 22, 2015). *Trap-Neuter-Return (TNR) Pilot*. `http://www.tinykittens.com/cats`.

Zillians. 2015 (accessed June 22, 2015). *CatFi - Smart Feeder*. `http://catfi.com/`.

Received February 1986; November 1993; accepted January 1996