Week 1: Constructive Research in Object Detection

Nate Bachmeier

TIM-7140:Software Engineering

April 25, 2021

Northcentral University

**Constructive Research in Object Detection**

Constructive design is one of the most common research methods for information systems and technology (Silvestrini & Sammito, 2012). These studies identify a problem, build solution artifacts, and communicate the implementation’s unique value (Hevner et al., 2004). For example, high-speed broadband internet is not available within many developing countries. These challenges promote researches to create new compression algorithms and improve the existing infrastructure. Typically, these results (artifacts) originate from specific Proof-Of-Concept (POCs) or directed case studies.

# Literature Review

Three constructive design articles from Northcentral University’s Library were selected (see Table 1). These articles identify a specific problem within object detection and then produce reusable artifacts (e.g., business processes and algorithms).

Table 1: Selected Papers

|  |  |  |  |
| --- | --- | --- | --- |
|  | Chebyshev Filter | Abandoned Objects | Low-Quality Images |
| Problem Statement |  | * Can AI detect garbage in the real world | * Can we detect people in low-quality camera feeds |
| Data Collection |  | * PETS 2006 * ABODA * Private Cameras | * Installed cameras in BSTU classrooms * Manually labeled training frames |
| Artifacts Produced |  | * Generic algorithm * Performance data | * Implementation in FCN, R-CNN, and Mask R-CNN |
| Contributions |  | * A mechanism for detecting garbage w.r.t. temporal and spatial rules | * A mechanism for counting people in low-resolution video * Assessed AI algo. performance |

## Detecting Humans on Low-Quality Images (2019)

Yudin et al. (2019) state that businesses require accurate counts of humans for various analytical processes. They address this problem with three deep-learning architectures (see Table 2) that can detect heads regardless of orientation, partial obstruction, and size (10x10 pixel minimum). Next, a performance assessment of the three architectures demonstrates the trade-offs between accuracy and processing time.

Table 2: Tested Algorithms

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Precision | Recall | Mean Duration Time (secs) |
| FCN with clustering | 80.3% | 66.9% | 0.22 |
| Faster R-CNN | 95.7% | 93.8% | 0.60 |
| Mask R-CNN | 99.8% | 97.8% | 2.91 |

The authors began the experiment by installing cameras in their college classrooms. They collected one thousand full-color frames in 1280x720 resolution under various lighting conditions. Each frame was manually labeled and split into training and validation sets (50-50%). During the training process, each architecture outputs bounding boxes around the detected heads. This approach enables the researchers to confirm both the counts and reasoning behind them.

Yudin et al. stream the IP-cameras into a Windows 7 desktop run which performs the analysis every minute. Statistical information (e.g., runtime) and bounded-box output collect into an SQLite database after each run. According to the results, Mask R-CNN architectures deliver the most accurate results. Meanwhile, FCN is nearly ten times faster but comes with lower quality results. Businesses can use this information to decide which characteristics are most critical to their workloads.

## Detecting Abandoned Objects (2020)

Park et al. (2020) state that businesses spend significant resources discovering abandoned objects across their premises. Typically, they address these challenges by hiring staff to monitor security cameras. However, this approach is expensive, manually intensive, and error-prone. Existing artificial intelligence (AI) systems also encounter challenges maintaining abandoned objects in the foreground versus fading into the background. Engineers can overcome this issue through additional computations.

Unfortunately, low-powered cameras lack the necessary resources to execute those algorithms. The researchers solve these issues with a dual-background model supporting objects fading into a long-term and short-term buffer. They also include crude object detection with contour analysis and a hierarchical entity tracking mechanism. These capabilities power a custom rules engine that reduces false positives by accounting for temporal and spatial information.

Park et al. assessed the solutions performing with a combination of open-source videos and personal cameras. Their experiments demonstrate the effectiveness of this process in many everyday environments. However, they also note several edge cases (e.g., toggling light sources) that require additional considerations. The process is also broadly applicable to other object detection problems. Even with virtually unlimited resources (e.g., cloud computing), using the dual-background model reduces re-analyzing the entire frame. This characteristic lessens memory requirements and operational costs.

## Object Detection Using Chebyschvev Filter (2021)