

IDUNAS	NATURAL & APPLIED SCIENCES JOURNAL	2021 Vol. 4 No. 2 (38-52)
--------	---------------------------------------	------------------------------------

# BabyTube Ruby on Rails Based Automatic Video and Image Tagging Application

Research Article

Leyla Kapi Kurtul<sup>1\*</sup> , Meriç Çetin<sup>2\*</sup> 

<sup>1</sup>Ruby on Rails Developer at YFU-Deutschland Oberaltenallee 6, 22081 Hamburg, Germany.

<sup>2</sup>Department of Computer Engineering, Pamukkale University, Kınıklı Campus, Denizli, Turkey.

Author E-mails

leylakapi@gmail.com

mcetin@pau.edu.tr

\*Correspondance to: Leyla Kapi Kurtul, Ruby on Rails Developer at YFU-Deutschland Oberaltenallee 6, 22081 Hamburg, Germany.

DOI: 10.38061/idunas.850198

Received: 30.12.2020; Accepted: 09.07.2021

## Abstract

The importance of technology in our lives is increasing day by day. The ability of people to keep up with developing technology is related to their correct and effective use of technology. Although positive contributions are detected in the use of technology and internet interactions, especially for people in early childhood, negative effects may be observed depending on the duration of exposure. For instance, thanks to the opportunities provided by technology, people in early childhood contribute to their language, cognitive or psycho-motor development. In addition to these positive features, the widespread use of technological tools has extremely harmful effects on the environment and human health. According to pedagogues, children between the ages of 2 and 12 are consumers of image/video contents on websites. When the contents watched/seen is not under the supervision of the parents, the subliminal message to be given may affect the development of children negatively. For the stated reasons, it is obvious that the videos/images uploaded in the web environment should be filtered and a content editor is required. In this paper, a software was developed to prevent the negative aspects of technological interactions (computer, tablet, phone, internet) of people in early childhood. This developed software automatically tags the contents in the web environment with Ruby on Rails based video and image processing. Using automatic content tagging, personal content control can be performed by providing systematic control over the web 7/24. More than 90% accuracy performance was achieved thanks to the power of Ruby and the Clarifai service, which is based on computer vision methods. It is thought that this approach will contribute to strengthening the social communication of people in the early childhood and developing their imaginations. The accuracy of the Ruby on Rails based video and image processing application has been tested with various machine learning techniques in this study. Therefore, it can be said that the developed software will pave the way for children to be mentally, physically and sensually healthier. In this way, it will be possible

to prevent various diseases (dyslexia, asperger syndrome, autism), which have increased in recent years and have been stated by some sources as technology based.

**Keywords:** Ruby on Rails, Clarifai, video and image tagging, content categorization, web application.

## 1. INTRODUCTION

The early childhood covering the period from birth to primary school age is very important for child development and education (Kaçar and Doğan, 2007). At these periods, which are thought to be critical for child development, the individual acquires basic knowledge, skills and habits (Çevik et al., 2017). Their dream worlds and perceptions are highly developed, and at the same time they are active, curious and investigative (Şahin and Akman, 2018). The education should not be left to chance at the early childhood, and it should be carried out systematically.

Technology has been an indispensable part of human life by shaping the environment since its existence (Sayan, 2016, Ergüney, 2017). The importance of technology in our lives is increasing day by day. The ability of people to keep up with developing technology is related to their correct and effective use of technology. Besides, it should not be forgotten that technological devices and internet usage are just tool for us. Although positive contributions are detected in the use of technology and internet interactions, especially for people in early childhood, negative effects may also be observed depending on the duration of exposure (Güngör, 2014, Aral et al., 2018). For instance, thanks to the opportunities provided by technology, people in early childhood contribute to their language, cognitive or psycho-motor development.

It has been stated that the use of technology in early childhood supports abilities such as imagination, different and analytical thinking, problem solving, and creativity (Dilekmen and Bozan, 2014, Akgündüz and Akpınar, 2018). In addition to these positive features, the widespread use of technological tools has extremely harmful effects on the environment and human health. There are many studies in the literature investigating the effect of technology usage on family communication. In these studies, it has even been stated that excessive usage of technology may cause the factors that hold the family together to disappear. The members of family rely on technology to increase efficiency in their home, school, business and social life (Claxton et al., 2016). Children as well as adults in the family are exposed to the use of technology, voluntarily or unintentionally. Some studies indicate that an average of 7.5 hours of technological device and / or internet use per day in early childhood. It is observed that the mental development of the children is negatively affected due to the usage of technological devices such as tablets, PCs, phones or internet. According to pedagogues, children between the ages of 2 and 12 are consumers of image/video contents on websites. Although the content encountered is instructive, the subliminal message may adversely affect child development. When the contents watched/seen is not under the supervision of the parents, even worse results can occur. For the stated reasons, it is obvious that the videos/images uploaded in the web environment should be filtered and a content editor is required.

Automatic identification of media contents such as images or videos is very important for their analysis. Although low-level diagnostic processes were used in previous years, many advanced studies have been conducted in terms of content characterization, emotion recognition or interactive content generation in recent years (Kanishcheva and Sharonova, 2018, Li et al., 2019). Most of these studies are based on deep learning networks, image processing techniques and various artificial intelligence methods. In this paper, a software was developed to prevent the negative aspects of technological interactions (computer, tablet, phone, internet) of people in early childhood (Kapi, 2019). This developed software automatically tags the contents in the web environment with Ruby on Rails based video and image processing. Using automatic content tagging, personal content control can be performed by providing systematic control over the web 7/24. It is thought that this approach will contribute to strengthening the social communication of people in

the early childhood and developing their imaginations. The accuracy of the Ruby on Rails based video and image processing application has been tested with various machine learning techniques in this study. Therefore, it can be said that the developed software will pave the way for children to be mentally, physically and sensually healthier. In this way, it will be possible to prevent various diseases (dyslexia, asperger syndrome, autism), which have increased in recent years and have been stated by some sources as technology-based.

## 2. WHAT IS BABYTUBE?

The starting point of this study is based on the impact of technology on child health. BabyTube is a video sharing platform controlled 24/7 by systematic controls. Thanks to this application, internet interactions of individuals in early childhood are controlled and filtering of negative content is guaranteed. BabyTube, which is basically developed with Ruby on Rails, works in integration with a wide variety of technologies. In the future, the videos are planned to be stored in databases on Amazon Web Service (AWS) in the cloud. In this way, it is thought that this non-profit application can take advantage of the numerous features of AWS for free. The working principle of BabyTube is as follows:

- The contents uploaded to the system by the user are taken into the video/image processing process.
- The tags are created for the relevant video/image.
- Thanks to the control mechanism, the videos are filtered according to the produced tags.
- The videos are automatically separated according to the profile-based keywords determined by the parents as a result of the filtering.
- Finally, the related content is made accessible to everyone by the administrator.

Using the BabyTube application, parents get information about the content that their children access and give them freedom in their content choices. In addition, BabyTube application monitors and analyzes the user's movements thanks to AI (Artificial Intelligence) based system features.

## 3. BABYTUBE CODING AND WORKING STRUCTURE

In that section talking about technologies which are used in BabyTube, coding structure, and working style of application. BabyTube coding and working structure are gathered under three main headings and detailed.

### Technologies Which Are Used

The current technologies used in BabyTube application are as follows:

- HTML
- HAML
- CSS
- SASS
- Bootstrap
- Coffescript
- PostgreSQL
- Ruby
- Ruby on Rails
- NodeJS

- Clarifai API
- Yandex Metrica

Ruby is an open-source free software developed by Yukihiro Matsumoto (Flanagan, 2008). Ruby on Rails is a web framework developed by David Heinemeier Hansson (DHH) and written in the Ruby programming language (Bächle and Kirchberg, 2007). Ruby on Rails structure, which is generally used in web-based applications, contains all the components that applications need (Ruby on Rails Software, 2021). The strength of the Rails framework is that it is open source. In this way, it provides great flexibility and freedom to developers.

### **Coding Structure**

Tables and table relationships are shown in Figure 1 for developed BabyTube Ruby on Rails application. Rails model structure is that provides the relationship between data in database and packages. There are not only relationships but also connections in the structure. The explanation of these connections over the video and tag relationship in the system is shown in Figure 2. Besides the video model, the user model has also been added. The relationship between user and video is shown in Figure 3. There is no doubt these facilities which are in the model structure increase use of Ruby on Rails. In Figure 4, a representation from the model class is given. User ownership is added with the 'has\_many' relationship in the user model. In this way, the structure that enables the user to access videos and tags and the relationships in the database are added to the model's side. The use of the Rails web framework also increases thanks to the convenience in the model structure, such as the developer's application without open the database from within the model.

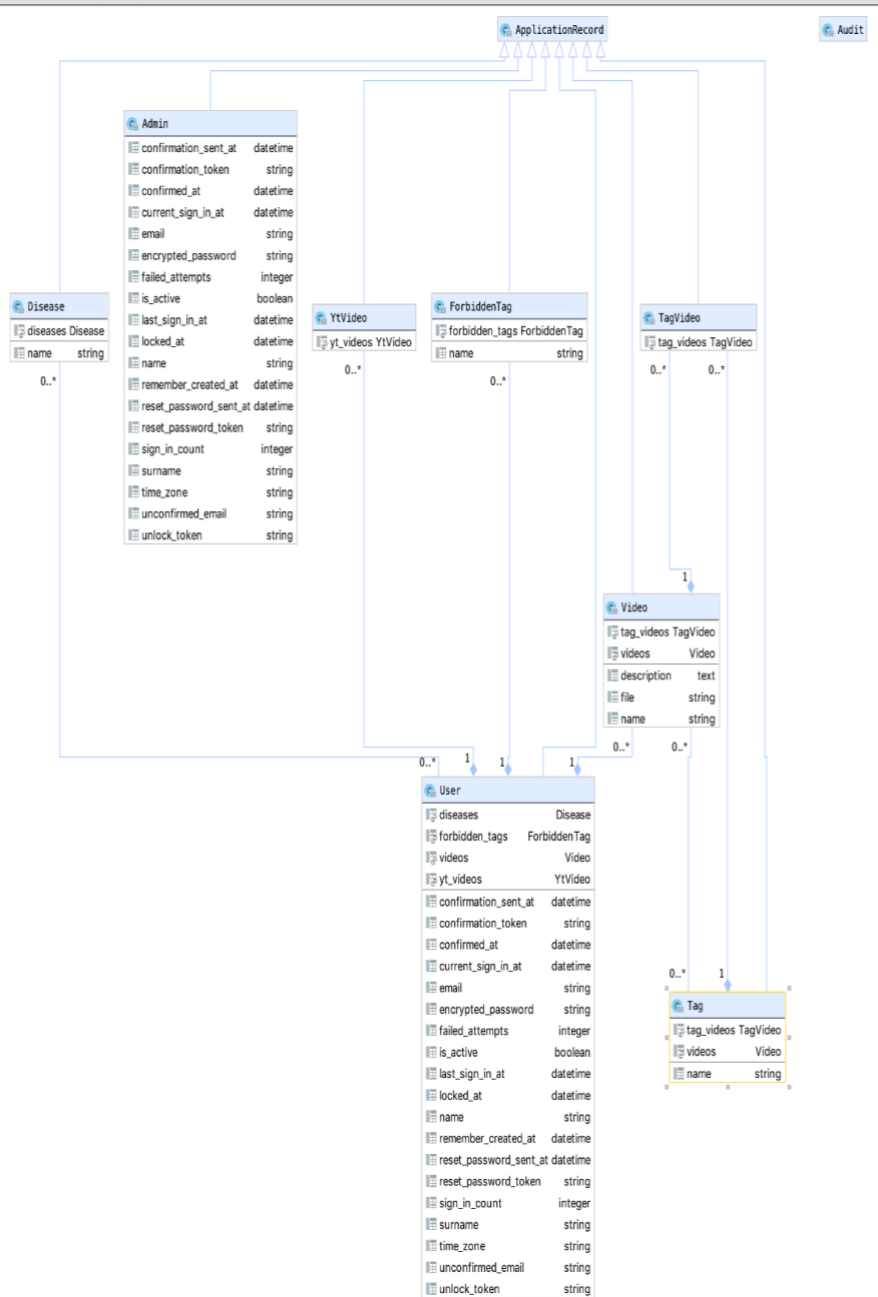


Figure 1. BabyTube database schema.

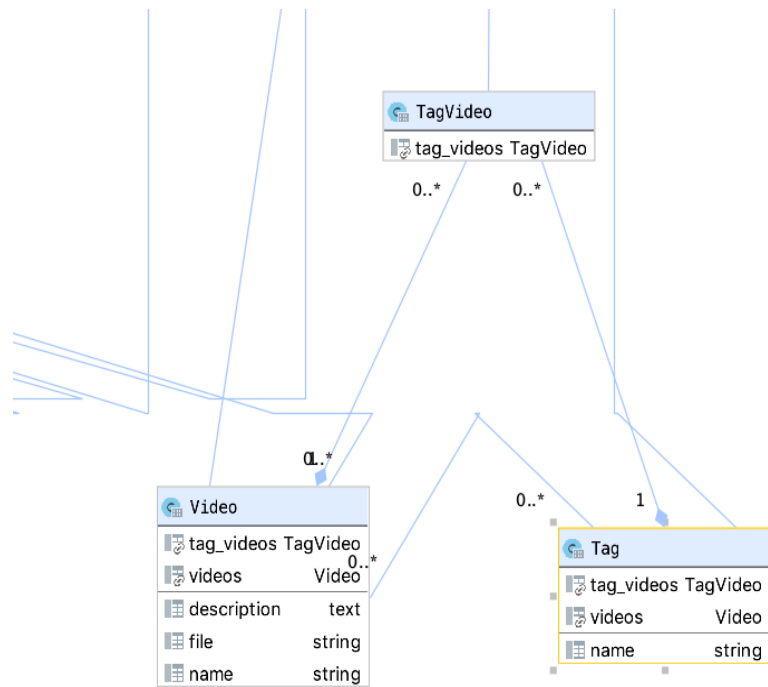


Figure 2. Database representation of the relationship structure between video and tag.

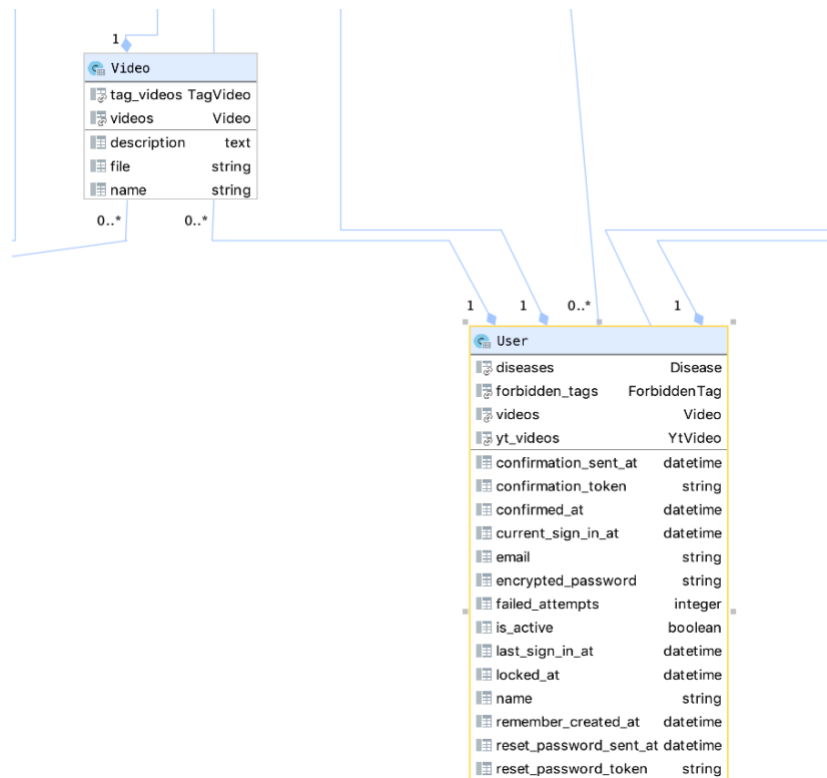


Figure 3. Database representation of the relationship structure between video and tag.

```

1 # frozen_string_literal: true
2
3 class User < ApplicationRecord
4   include PasswordCreatable
5
6   # Scopes
7   scope :active, => { where(is_active: true) }
8
9   # Include default devise modules. Others available are:
10  # :confirmable, :lockable, :timeoutable and :omniauthable
11  devise :database_authenticatable,
12         :registerable,
13         :recoverable,
14         :rememberable,
15         :trackable,
16         :validatable
17
18  # Send devise emails with background job
19  def send_devise_notification(notification, *args)
20    devise_mailer.send(notification, self, *args).deliver_later
21  end
22
23  # Relations
24  has_and_belongs_to_many :diseases, name: :diseases
25  accepts_nested_attributes_for :diseases
26
27  has_many :videos
28  has_many :yt_videos
29  has_many :forbidden_tags
30  # Helpers
31  audited except: [:password]
32

```

Figure 4. The representation of database from model class.

### Controller Structure

Model-View-Controller (MVC) is an important framework that includes a View-Controller (VC) framework. In MVC, Action Controller handles the user interaction and carries out the control task of the structure. Basically, session management, template creating and redirecting. Action View is responsible for template management and provides control of the views displayed on the screen. This layer has many technologies supports. The last part is Action Dispatch. It sends the relevant request to the relevant links according to the structure of the requests. The structure that should be executed on the terminal for the control structure in the application is the "rails g controller controller\_name" command. By using "rails generate controller video" command in terminal, the video control structure is added as shown in Figure 5.

```

1 class Users::VideosController < Users::ApplicationController
2
3   before_action :set_video, only: [:show, :edit, :update, :destroy]
4   add_breadcrumb "VideoLar", path: user_videos_path
5
6   def index
7     if params[:tag].present?
8       @search = current_user.videos.joins(:tags).order(id: :desc).search(tags: { name: params[:tag] })
9     else
10      @search = current_user.videos.order(id: :desc).search(params[:q])
11    end
12    @videos = @search.result(distinct: true).paginate(page: params[:page])
13    respond_with(@videos)
14  end
15
16  def new
17    @video = Video.new
18  end
19
20  def create
21    @video = Video.new(video_params)
22    @video.user_id = current_user.id
23    if @video.save
24      params['video']['tags'].uniq!.each do |tag|
25        new_tag = Tag.find_by_name(tag)
26        TagVideo.create(tag_id: new_tag.id, video_id: @video.id)
27      end
28      render json: @video
29    else
30      flash[:error] = @video.errors.full_messages
31    end
32  end
33
34  def show
35    add_breadcrumb @video.name, user_video_path(@video)
36    respond_with(@video)
37  end
38
39  def edit
40  end
41
42  def update
43    @video.update(params_video)
44    respond_with(:user, @video)
45  end
46
47  def destroy
48    if @video.tags.present?
49      @video.tags.delete_all
50    end
51    @video.destroy
52    respond_with(:user, @video)
53  end
54

```

Figure 5. Video control structure.

### Router Configuration

The router in Rails is a router module that recognizes browser URLs and sends them to the requested controller actions. When a URL is entered into the domain, the Rails router structure knows which controller

and action will handle the URL. Routers also work with incoming control structures and show the outputs to the end user. Figure 6 shows the Rails route structure schematically.

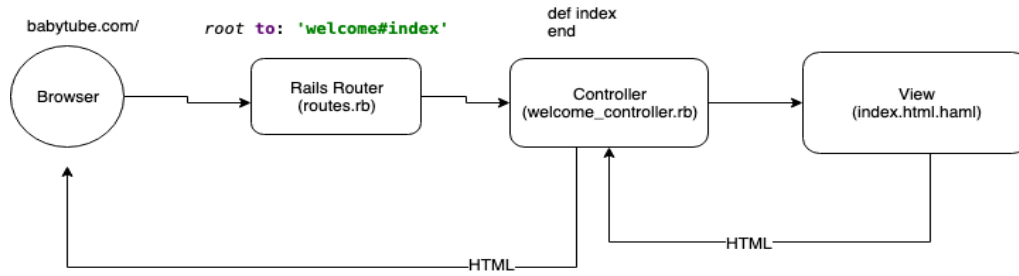


Figure 6. Rails route structure.

### Gems Structure

A "Gemfile" is a file used in Ruby programs to define "gem" dependencies. A "Gem" is a collection of code made up of many Ruby files that can be used later. The "Gemfile" file should always be in the root directory of the project. In general, Ruby on Rails can add desired features very easily with its "Gem" structure.

Some of the "Gem" structures of Ruby on Rails (ROR) application can be summarized as follows:

- i) The "Devise Gem" is one of the most popular libraries for authenticating in Rails applications.
- ii) The "Simple-form Gem" is a library that is used to improve the form structures in the system and make the forms perfect with powerful components. Thanks to the "Simple-form Gem" structure, the code quality increases, and a cleaner code structure is offered.
- iii) The "YT Gem" provides integration of YouTube videos to the system. This feature not only brings the videos but also the features of the videos.
- iv) By using "Carrierwave gem", image and video structure is added to the system. In this way, video and image upload is provided in a very easy and practical way. In addition, it provides the opportunity to store the files in the desired format and size, so that the same file can be displayed, stored and worked on in different formats.

### Working Structure and Clarifai API

Clarifai service is a popular third-party service that makes inferences from images, videos and text using computer vision and natural language processing in a one integrated AI Computer Vision platform (Clarifai, Technology, 2021). Clarifai is a machine learning method which utilizes CNN-Convolutional Neural Networks and DNN-Deep Neural Networks to analysis processes. Machine CORE (Clarifai Object Recognition Engine) is behind this learning method, thus, it is possible to test millions of images with this system. A comparison of Clarifai with leading API providers is shown in Table 1.



**Table 1.** Comparing the features of Clarifai and other tools

Features	Amazon	Google	Clarifai	Microsoft
Image tagging	Yes	Yes	Yes	Yes
Video tagging	Yes	Yes	Yes	Yes
Emotion detection	Yes	Yes	Yes	Yes
Logo detection	Yes	Yes	Yes	Yes
NSFW tagging	Yes	Yes	Yes	Yes
Dominant color detection	Yes	Yes	Yes	Yes
Feedback API	No	Yes	Yes	No
Price (per 1000 image)	1\$	1.50\$	1.20\$	1.50\$
Free request right	5000/Month	1000/Month	5000/Month	5000/Month
Image size limit	5Mb/Image	20Mb/Image	Undefined	4Mb/Image

Clarifai API, which can work in integration with many programming languages, is a more effective and easy service compared to its equivalents. An example of image analysis with Clarifai API is as: In web application, 100 default requests were made to the Clarifai service for both tested images. The obtained simulation results were published as good, good and bad (irrelevant). The images presented in Figure 7 were tested by Clarifai service with colored pages.



**Figure 7.** Croissant and Main Coon Cat images.

In 100 requests for Croissant: The best result was found at 98% level as "Croissant". Good level 96% "Bread", 93% "Dough" and 87% "Sugar" returned. At the worst level, 83% returned an irrelevant label as "Cooling". 100 requests in the Main Coon Cat Image: As 100% "Cat" at the best level returned. At a good level 99% "Cute", 96% "Purebred" and 94% "Furry" returned. It returned a tag that was 94% "Young" at a poor level. To explain in detail, the calls for croissant painting the croissant returned at 98, and this is the best match.

## How Application is Working?

In the application, the user uploads the desired videos to the system after logging in to the page. As a result of the application, the tags in the video analysed are obtained. These tags are used later for search and filter operations. In addition, when the "video id" information is given as input, videos suitable for the content can be presented to the user via the "Youtube" application.

After register or signed into the application, by using the "New Video" button, the mp4 file can be uploaded to the system and a new video can be added for analysis. While the video is uploaded to the system, NodeJS starts to analyze the video. Afterwards, with a "post" request, the tags are saved in the controller structure. The tags returned as a result of the "POST" request are shown in Figure 8. A different "post" request is sent to save the returned tags to the database.

As an example, Figure 8 shows the returned tags and accuracy percentages for a 5-second video (https://www.youtube.com/watch?v=EngW7tLk6R8) added to the application. The application returns many tags as a result of video analysis and these tags are listed as those with an accuracy value greater than "0.90".

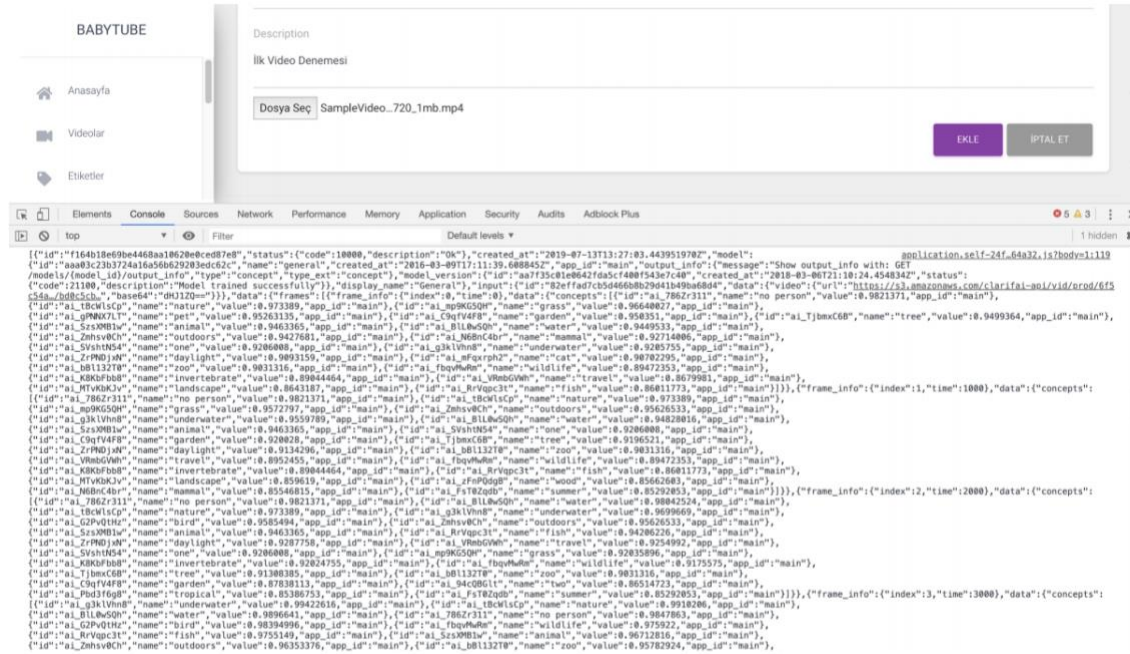


Figure 8. Tags that return at the end of the analysis after video is uploaded.

A different "POST" request is sent to save the returned tags to the database. The result of this post request is shown in Figure 9. Using this application, a comparison can be made from the tags returned from the uploaded and analysed videos. Additionally, videos containing user-specified prohibited words can be blocked from uploading. For example, the forbidden word is "tree". The code block required for content blocking is shown in Figure 10.

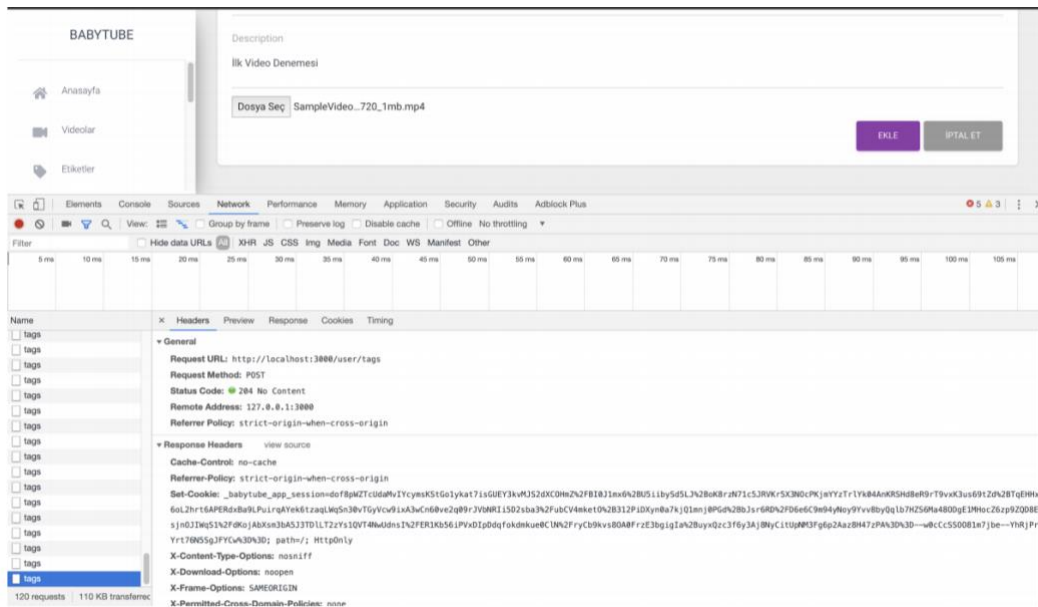


Figure 9. The analysed tags are sent to be saved in the Tag table with POST request.

If a video with a forbidden word tag is tried to be uploaded to the system, a message like this " The video cannot be added to the database." is generated. Thus, as a result of the analysed video, it is checked whether there is a blocked tag in the returned tags, and then the video is not recorded. Checking the database shows that the video is not recorded.

After upload video to the Babytube application besides of the blocked tags also can make CRUD operations. Recorded video can be edited and re-analysed when needed can be done. In addition, with this application, the video can be analysed by entering the 'video\_id' information of the Youtube video, as well as new videos suitable for the content to be associated with the 'video\_id' information can be presented to the user.

In the Figure 11 you can see a sample Youtube video to copy video\_id from URL. Figure 12 shows how Youtube 'video\_id' is entered into the system. After the 'video\_id' is transferred to the system, the tags of the video are generated. Figure 13 shows how the tags obtained from video analysis with Clarifai application appear in the system. In addition, the tags produced for the sample videos presented in Figure 13 and the accuracy values of these tags are listed in Table 2 and Table 3.

```

34 const not_allowed_word = "tree";
35 console.log(app);
36 var ready = function(){
37   $('#table').footable();
38
39   $('#datetimepicker').datetimepicker({
40     format: $('#datetimepicker').data('format'),
41     step: 60,
42     lang: 'tr',
43     minDate: 0,
44     defaultTime: '12:00'
45   });
46
47   $(function () {
48     $('[data-toggle="tooltip"]').tooltip();
49   });
50
51   $('[data-mask]').each(function(index, element) {
52     var element;
53     element = $(element);
54     element.mask(element.data('mask'));
55   });
56
57   $('#form[data-turboform]').on('submit', function() {
58     TurboLinks.visit( ( this.action + (this.action.indexOf('?') === -1 ? '?' : '&') + $(this).serialize() );
59     return false;
60   });
61
62   function findNameInArr(obj, findName) {
63     for (i = 0; i < obj.length; i++) {
64       for(j=0; j < obj[i]['data']['concepts'].length; j++){
65         result.push(obj[i]['data']['concepts'][j]['name']);
66       }
67     }
68     if(result.includes(findName)){
69       alert('not allowed to database saving');
70     }else{
71       alert('going to save');
72       //save to db to the tags
73       result.forEach(function(ref){
74         var formData = new FormData();
75         formData.append( name: 'tag[name]', ref);
76         console.log(formData);
77         $.ajax({
78           url: '/user/tags',
79           data: formData,
80           cache: false,
81           contentType: false,
82           processData: false,
83           dataType: 'json',
84           method: 'POST',
85           success: function (data) {
86             console.log(data);
87           }
88         });
89       });
90     }
91   }
92 }

```

Figure 10. Blocked tags code structure.

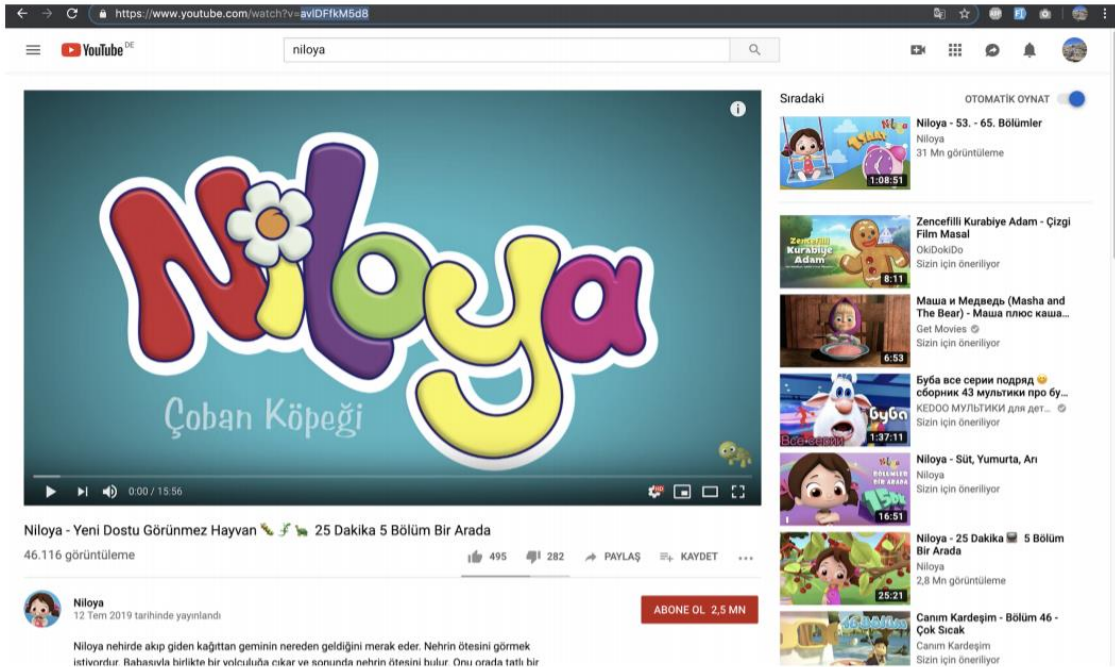


Figure 11. Sample Youtube video and video\_id taking process.

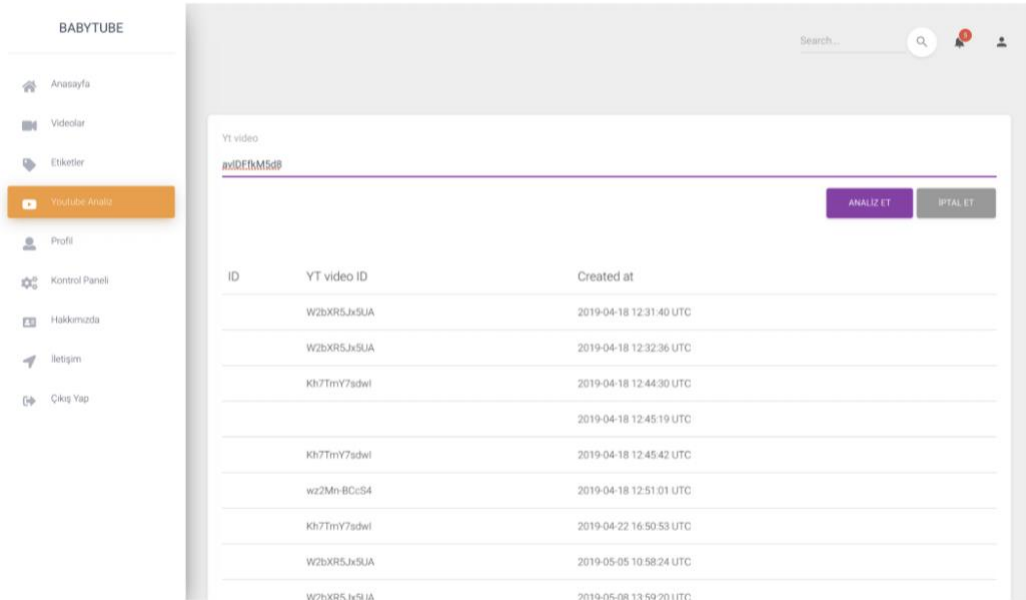


Figure 12. The process of copying Youtube video\_id to the system.

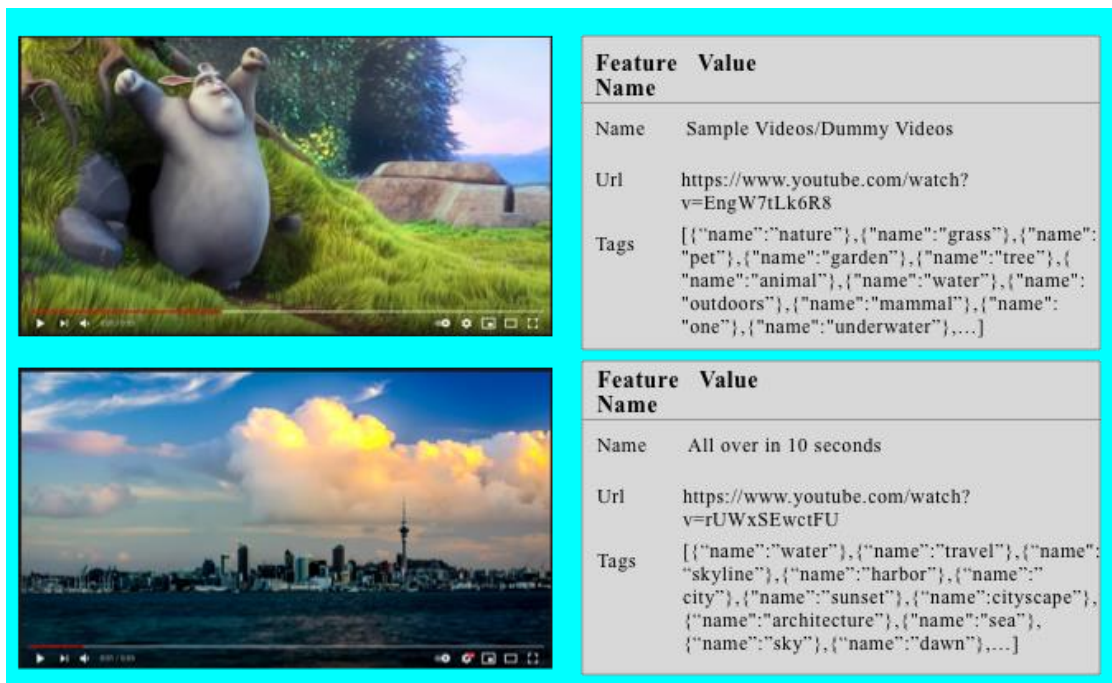


Figure 13. Produced tags of sample Youtube videos.

The analysis of sample videos was completed in an average of 2.32 seconds. The analysis size of the tags is 2.4 KB. In Hamburg, Germany location, the connection speed for video analysis: PING, 7 ms, download speed: 47.94 Mbps, upload speed: 21.95 Mbps. When these values are evaluated in terms of both accuracy and speed, they clearly reveal the power of the system. According to the results of video/image analysis, not only objects but also emotions and qualities can be detected in the system. In this paper, which was done by combining the power of Ruby and Clarifai, when the bad results were removed, an accuracy rate of more than 90% was obtained. The results showed how compatible, healthy, and efficient Ruby, Ruby on Rails and Clarifai technologies work together. The system, which is being developed, will be able to produce more complex images and high accuracy results from different angles of view over time.

**Table 2.** The generated tags of sample video 1 with value > = 86

Tag	Value	Tag	Value
No person	0.98213712	One	0.92060080
Nature	0.97338900	Underwater	0.92057510
Grass	0.96640030	Daylight	0.90931606
Pet	0.95263130	Cat	0.90702266
Garden	0.95035094	Zoo	0.90313100
Tree	0.94993630	Wildlife	0.89472330
Animal	0.94633630	Invertebrate	0.89044440
Water	0.94495314	Travel	0.86799794
Outdoors	0.94276816	Landscape	0.86431860
Mammal	0.92714000	Fish	0.86011720

**Table 3.** The generated tags of sample video 2 with value > = 87

Tag	Value	Tag	Value
Water	0.99300100	Dawn	0.96019346
City	0.98977834	Building	0.95924324
Sunset	0.98438900	Downtown	0.95898900
Travel	0.98221534	Pier	0.93812925
Cityscape	0.97174100	Skyscraper	0.91940820
Skyline	0.97047790	Ocean	0.91559740
Architecture	0.96983970	Urban	0.89735160
Harbor	0.96943890	Dusk	0.88911676
Sea	0.96850430	Watercraft	0.88071500
Sky	0.96412070	Port	0.87772230

**4. RESULTS AND RECOMMENDATIONS FOR FUTURE STUDIES**

In this study, a Ruby on Rails based application has been developed to make technological device and internet usage of children safer and healthier. This developed software automatically tags the contents in the web environment with Ruby on Rails based video and image processing. Using automatic content tagging, personal content control can be performed by providing systematic control over the web 7/24. It is thought that this approach will contribute to strengthening the social communication of people in the early childhood and developing their imaginations. The accuracy of the Ruby on Rails based video and image processing application has been tested with various machine learning techniques in this study. Therefore, it can be said that the developed software will pave the way for children to be mentally, physically and sensually healthier. The improvable parts of the BabyTube application can be listed as follows:

- With an integration to be added to the application, Youtube videos can be analyzed systematically.
- Authorization can be given to pedagogues to interact.
- The "Disorders" part, which is present in the infrastructure but not visible at the interface, can be detailed and the video ranking can be done according to this menu.
- Different analyzes can be made with the generated labels and their graphical representation can be provided
- A timer for children's screen usage can be added.
- Using Metrica, features such as user mood, disorder detection can be added as a result of the analysis of mouse movements.

## 5. ACKNOWLEDGEMENTS

Abstract version of this paper was presented at International IDU Engineering Symposium – IES’20, December 5-6 and 10-13, 2020, Izmir, Turkey with the title of “Ruby on Rails Based Automatic Video and Image Tagging Application”.

## 6. REFERENCES

1. Akgündüz, D., & Akpınar, B. C. (2018). Okul Öncesi Eğitiminde Fen Eğitimi Temelinde Gerçekleştirilen STEM Uygulamalarının Öğrenci, Öğretmen ve Veli Açısından Değerlendirilmesi.
2. Aral, N., & Doğan Keskin, A. (2018). Ebeveyn bakış açısıyla 0-6 yaş döneminde teknolojik alet kullanımının incelenmesi. *Addicta: The Turkish Journal on Addictions*, 5(2), 317-348.
3. Bächle, M., & Kirchberg, P. (2007). Ruby on rails. *IEEE software*, 24(6), 105-108.
4. Claxton, G., Rae, M., Long, M., Damico, A., Whitmore, H., & Foster, G. (2016). Health benefits in 2016: family premiums rose modestly, and offer rates remained stable. *Health Affairs*, 35(10), 1908-1917.
5. Clarifai, Technology, (2021). Available at: <https://www.clarifai.com/technology> (retrieved March 13, 2021).
6. Çevik, G., Yılmaz, R. M., Goktas, Y., & Gülcü, A. (2017). Okul Öncesi Dönemde Artırılmış Gerçeklikle İngilizce Öğrenme. *Öğretim Teknolojileri Ve Öğretmen Eğitimi Dergisi*, 6(2), 50-57.
7. Dilekmen, M., & Bozan, N. (2014). Okul Öncesi Eğitimde Oyunun Öğretmen Görüşlerine Göre Değerlendirilmesi. *Atatürk Üniversitesi Kazım Karabekir Eğitim Fakültesi Dergisi*, (37), 43-56.
8. Ergüney, M. (2017). İnternetin Okul Öncesi Dönemdeki Çocuklar Üzerindeki Etkileri Hakkında Bir Araştırma. *Ulakbilge Sosyal Bilimler Dergisi*, 5(17), 1917-1938.
9. Güngör, M. (2014). Okulöncesi Dönem Çocuklarının Televizyon İzleme Alışkanlıkları Ve Anne Baba Tutumları. *Mustafa Kemal Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 11(28), 199-216.
10. Flanagan, D., & Matsumoto, Y. (2008). *The Ruby Programming Language: Everything You Need to Know*. " O'Reilly Media, Inc."
11. Kacar, A. Ö., & Doğan, N. (2007). Okul öncesi eğitimde bilgisayar destekli eğitimin rolü. *Akademik Bilişim*, 31, 1-11.
12. Kanishcheva, O., & Sharonova, N. (2018). Image and Video Tag Aggregation. *In AIST* (pp. 161-172).
13. Kapi, L. (2019). Erken çocukluk dönemi gelişimi için ruby on rails temelli video ve resim işleme uygulaması: Babytube (Master's thesis, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü).
14. Li, X., Xu, C., Wang, X., Lan, W., Jia, Z., Yang, G., & Xu, J. (2019). COCO-CN for cross-lingual image tagging, captioning, and retrieval. *IEEE Transactions on Multimedia*, 21(9), 2347-2360.
15. Ruby on Rails Software, (2021). Available at: <https://rubyonrails.org/> (retrieved March 13, 2021).
16. Sayan, H. (2016). Okul Öncesi Eğitimde Teknoloji Kullanımı. *21. Yüzyılda Eğitim Ve Toplum Eğitim Bilimleri Ve Sosyal Araştırmalar Dergisi*, 5(13).
17. Şahin, M. K., & Akman, B. (2018). Erken Çocukluk Döneminde Düşünme Becerilerinin Gelişimi. *Milli Eğitim Dergisi*, 47(218), 5-20.