

ARTranslate - Immersive Language Exploration with Object Recognition and Augmented Reality

Benny Platte¹, Anett Platte², Rico Thomanek¹, Christian Roschke¹,
Tony Rolletschke¹, Frank Zimmer¹, Marc Ritter¹

¹ Technikumplatz 17, 09648 Mittweida, platte@hs-mittweida.de

² Marschnerstraße 31, 04109 Leipzig

Abstract

The use of *Augmented Reality* in teaching and learning contexts for language is still young. The ideas are endless, the concrete educational offers available emerge only gradually. Educational opportunities that were unthinkable a few years ago are now feasible.

We present a concrete realization: an executable application for mobile devices with which users can explore their environment interactively in different languages. The software recognizes up to 1000 objects in the user's environment using a deep learning method based on *Convolutional Neural Networks (CNN)* and names this objects accordingly. Using *Augmented Reality (AR)* the objects are superimposed with 3D information in different languages. By switching the languages, the user is able to interactively discover his surrounding everyday items in all languages.

The application is available as Open Source.

Keywords: AR, Augmented Reality, immersive learning, language learning, deep learning, object recognition, SceneKit, CoreML

1. Introduction

Language is the foundation of good communication and “one of the most important prerequisites and foundations for culture”. (Beller and Bender, 2010, p. 169). Language acquisition is regarded as the key to social integration

Just as we only learn to swim in water, we can only learn to speak by speaking. Therefore there is no way around the practical usage. But how do we start?

Intrinsic motivation “for its own sake, independent of factors outside the person” (Brandstätter et al., 2013) is the best one. It has to be used. Let's take something almost everyone has with them: the smartphone (Oloruntoba, 2019). 75 % of 10-11 year olds in Germany own a smartphone. In the 12-13-year-old group there are even 95 % (Tenzer, 2019), in Switzerland 93 % (Schultz, 2018).

We absorb our mother tongue immersively. Learning a non-parental second language requires effort and willingness to learn. The willingness to learn is greater if

- we learn a second language as *ambient language* (Ruberger et al., 2017) like for example refugees in a foreign country. “In the conditions of migration the mastery of the language of the environment marks an important step on the way of integration.” (Adler, 2011, p. 104).
- the hurdle of getting us up is lowered by the simplicity of the learning context

This is exactly where our contribution comes in. We lower the hurdle of engaging in a new language by using the technology of *Augmented Reality (AR)* to enrich the user's environment with educational content (Fig. 1).

Today the use of AR technology is becoming increasingly popular in engineering (Ismail et al., 2019; Rana and Patel, 2019), environmental sciences (Barreiros et al., 2016) and especially education (Sirakaya and Alsancak Sirakaya, 2018; Kucuk et al., 2014)

Augmented Reality (AR) can be defined as a real world



Figure 1: *ARTranslate* in Action: Exploring the everyday World around with Augmented Reality (AR) The virtual objects “stick” to the real objects regardless of the viewing angle.

Online Resource with Source Code

<https://github.com/benpla/ARTranslate>

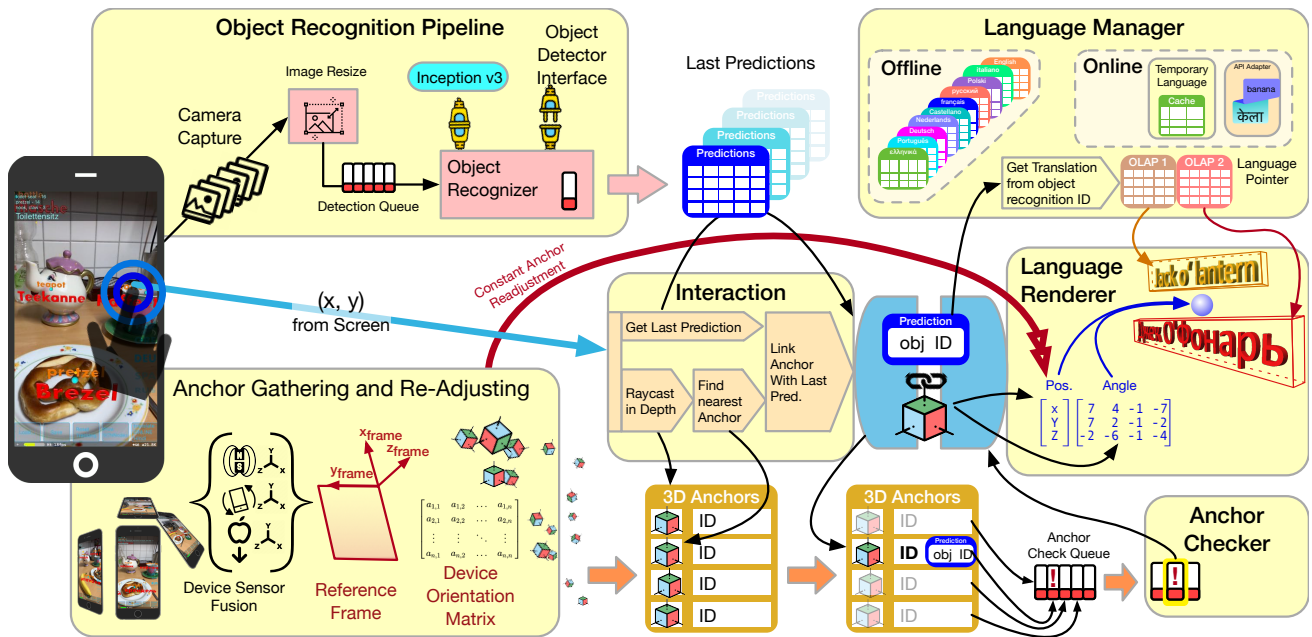


Figure 2: ARTranslate System Overview

context that is dynamically overlaid with context sensitive virtual information. AR has two main characteristics:

- real and virtual objects are overlaid
- interactivity

In a AR interface, users are viewing the world through a portable display that overlays graphics on videos of the environment. *Augmented Reality* Interfaces allow a person to interact with the real world in a new way. The user can move through the three-dimensional virtual room and view it from any point, just like a real object.

In the context of language education by means of AR new territory is emerging recently. The first providers are on the market. Ideas and concepts have to be explored and evaluated.

For this purpose we present *ARTranslate*, a mobile application that enriches the user's environment with new languages (Fig. 1). The source for this is an object recognition algorithm that uses *Convolutional Neural Networks (CNN)* to recognize, name and display the objects in the environment in many languages. The visualization is three-dimensional and moves with the user, the overlaid information "sticks" to the objects surrounding the user.

1.1. Related Work

Previous work on language acquisition with the help of *Augmented Reality (AR)* are few and far between.

Marker Based AR Learning via AR is used in many places. There are also language learning apps that use this technology. An example for marker-based AR is the app "MySmart Flashcards". However, this does not even include layers. A virtual object is only placed on a maximum of 50 previously learned (and purchased) cards, which must be in the camera image (Bobrovskaya, 2019). The augmentation of prefabricated cards with voice information also investigated Dicky et al (Dicky and Chuah, 2019). A very nice app is mARble

(MedAppLab, 2019). Persons stick small printed markers on the skin. The app then shows forensic injuries. Students of forensic medicine can learn with it.

Markerless AR In this case, the camera recognizes the contour or structure of an object, which triggers an event such as the insertion of information. The company Mondly (ATI, 2019) is one development step further. They advertise with "AR-supported learning". It searches for planes in space and places a virtual assistant on them who presents different things. An interaction with the environment here is only rudimentary: the objects in the environment are not included in the AR Experience.

2. Application Design

ARTranslate is a software implementation based on Apple iOS mobile devices. As IDE XCode in version 11.1 was used. The software runs on iOS from version 12 and is available at the URL listed on the title page below.

ARTranslate is based on four pillars:

- Capture the camera image and feed it into a high frequency deep learning object recognition algorithm for the identification of things within the image.
- Follow the devices orientation and movement in space. Capture the space coordinate system and collect anchor points.
- interaction controller. During user interaction, the objects are anchored in the room.
- A language manager delivers the texts either offline (for some languages, expandable) or online ("more than 100 languages" (Wu et al., 2016)).
- The Language Renderer calculates the 3D representation of the texts and displays them at the anchor points using the AR algorithm.

2.1. Object Recognition: Prediction Pipeline

Image recognition is an area of application for neural networks. *ARTranslate* provides an interface for docking different image recognition algorithms. The modular implementation allows an easy exchange of the neural model later on. We decided to use the deep learning model Inception-v3 (Szegedy et al., 2016). The Inception-v3 network has 24 million parameters. The pre-trained net was trained with the ImageNet image set. The validation and test data will consist of 150,000 photographs, collected from flickr and other search engines, hand labeled with the presence or absence of 1000 object categories (Russakovsky et al., 2017).

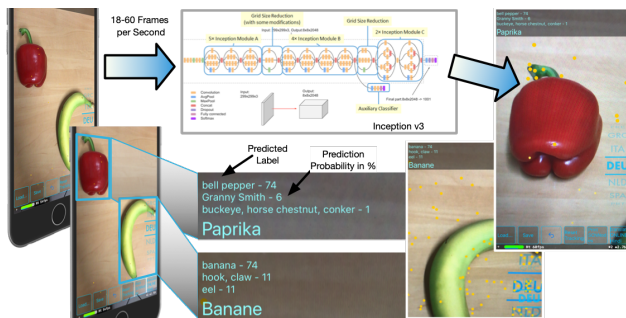


Figure 3: Object Detection with Deep CNN Inception v3

The object recognition interface was implemented using Core ML (Apple, 2019). The camera image is being captured in real time. Up to 60 frames per second (60 fps) are transferred to the Object Recognition Module. In the first step, the Object Recognition Module downsamples the images to a lower resolution of 299x299 pixels. Then the reduced image is transferred to a processing queue together with an asynchronous callback function. Then the next image is captured, resized and put into the queue. The attached object classification model reads the requested images from the input queue and returns a list of predictions. These contain the recognized objects with their respective probabilities. The object classifier calls the callback function passed with the requested prediction. This function stores the last real-time prediction list, displayed in “Last Predictions” in Fig. 2 at the top right.

By passing the asynchronous callback function, each of the system modules works at its own speed. Individual longer runtimes are compensated. The user interface does not have to wait for every prediction. The condition for this is that the performance of the mobile device is sufficient to process all operations *on average*.

This workflow runs continuously with the start of the software.

2.2. Anchor Gathering

While the prediction pipeline is running, the system searches for anchor points in the room. Anchor points are invisible null-objects that can hold a 3D content in World Space. For orientation in 3D space, the exact orientation of the device in the room is required. This orientation is determined by three sensors:

- Accelerometer measures proper acceleration relative

to freefall, felt by people or objects. Units: $\frac{m}{s^2}$ or g . iPhone 4 value range: $\pm 2g$, precision 0.018g

- Gyroscope detects the current orientation of the device or changes in the orientation. The output are 3 angles: Yaw (rotation around z axis), Pitch (around x axis) and Roll (around y axis), expressed in $\frac{rad}{s}$.
- Magnetometer measures the strength of earth’s magnetic field, strength is expressed in tesla [T]. iPhone 4 magnetometer range: $\pm 2mT$

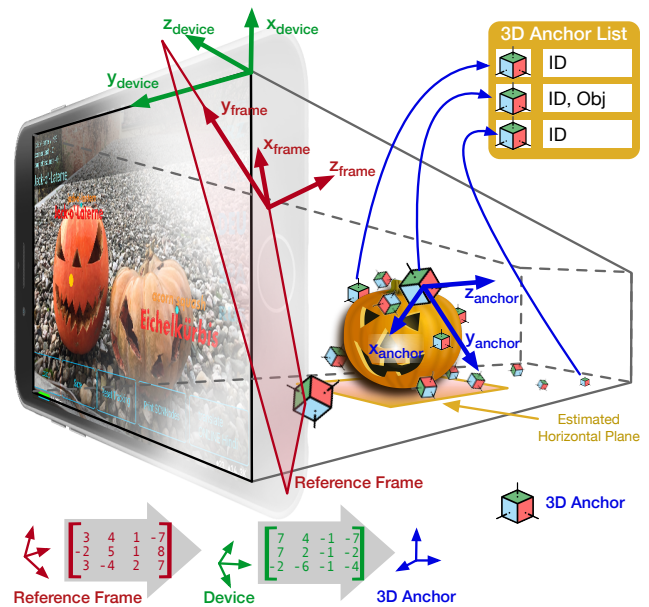


Figure 4: Anchor Gathering

The magnetic field is assumed to be almost fixed at the time of data recording. The user is what moves. First, the coordinate systems of the sensors installed in the smartphone are rotated into the outer world. The framework determines the current orientation in space in the format of a rotation matrix. A Reference Frame was chosen for *ARTranslate*, in which the Z axis is vertical and the X axis points toward true north. The *True North Reference Frame* requires the location in order to calculate the the declination of apparent magnetic to true north and the difference between magnetic north and true north (Hopping from apparent magnetic north to the true magnetic north further to the true north). The rotation matrices of the individual hops are concatenated and form a resulting total rotation matrix, which contains the rotation of the device coordinate system in all axes in relation to the Reference Frame.

With knowledge of the spatial orientation, anchor points in the captured 3D space will be searched in parallel. The anchor points will be set at relevant edges in the camera image. Their positions in the 3D world coordinate space are determined by changing the device orientation.

The anchor points are tagged with identifiers, temporarily stored and continuously adjusted. The world tracking becomes more accurate as the duration of the acquisition grows. With each improvement in space geometry, the anchor points that have already been temporarily stored are readjusted so that virtual objects move closer to the real world. In the

background, a point cloud full of temporarily stored anchor points is created in this way.

2.3. Interaction: Translation Object Positioning

By tapping on a location within the screen, the user requests interaction. At this moment, the system is in the following state:

- The object classifications of the last frames are buffered as lists and are available.
- A space coordinate system with anchor points exists. The accuracy does not play a role yet, because the world coordinate system is continuously adapted.

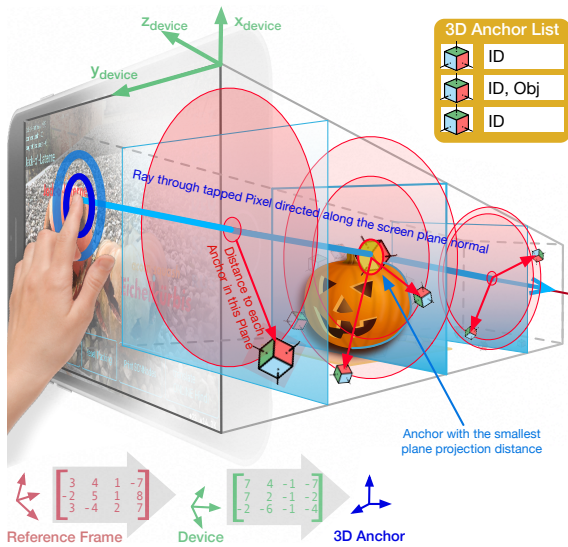


Figure 5: Interaction Stage: The search for the right anchor inside Language Renderer

The user typed *now* on the screen (Fig. 5). He sees the current camera image and his typing refers to an object which he sees *in the 2D image*. The anchor manager uses the screen coordinates of the user tap to calculate which of the buffered 3D anchor points in the deep projection planes of the screen (virtual into the display) has the smallest distance to the user tap. The user only sees the screen. The user's interaction is thus interpreted in terms of this projection. The determined anchor is provided with a marker and a prediction identifier. The anchors contain the markers themselves and the current prediction at the time of the user's interaction. The flagged anchor is passed to the anchor check queue. This works asynchronously in order to smooth load peaks.

The anchor checker retrieves *all* Anchors from the anchor check queue. Anchors without markings are discarded. If the anchor checker receives a marked anchor, it passes it on to the Language Renderer.

2.4. Language Manager

In parallel to the above processes, the Language Manager works in the user interface process. The Language Manager in Fig. 6 doesn't know any object classes and doesn't know any anchors. The Language Manager is responsible for managing the languages.

In the user interface a control for the language setting is located. This control is managed by the language manager.

It links the corresponding control of the user interface with the language data Source. If the user modifies the language control, the language manager receives a message for changing the language. There are two variants for the further procedure: offline or online.

2.4.1. Offline Language Resource

Some languages are available offline to improve the user experience. The object recognition is executed several times per second. The retrieval of previously saved translations for numerous objects in the image is faster and saves resources. If the language manager receives the request to change the target language via its offline channel, an internal pointer is moved to a Destination Language Array. For each language to be displayed in offline mode, there is a *Offline Language Array (OLA)* provided with the application which contains the names of all object classes of the object classifier. Currently the arrays contain 1000 object names in 10 languages, listed in Table 1.

Language	Kürzel	Language	Kürzel
English	ENG	Dutch	NLA
French	FRA	Russian	RUS
German	GER	Polish	POL
Greek	GRC	Italian	ITA
Portuguese	POR	Spanish	SPN

Table 1: verfügbare Sprachen mit Sprachkürzel nach ISO 639-1

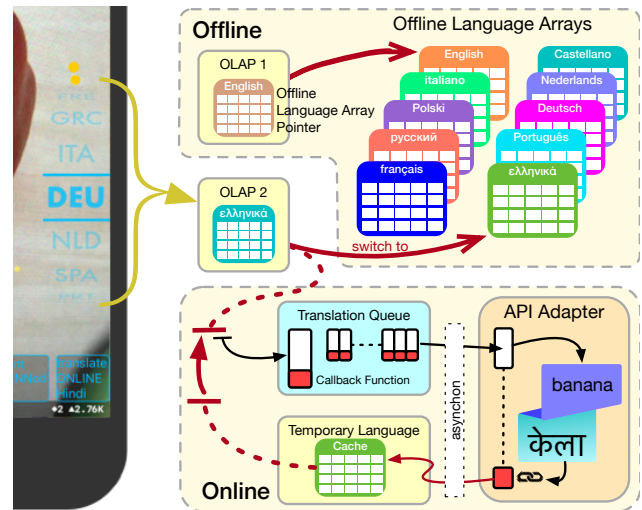


Figure 6: Language Manager with offline and online Language Channel

If a language switch is triggered by the language control, the Offline Language Array Pointer (OLAP) for the language 2 is set to the *OLA* of the desired target language. The Language Renderer has no knowledge of languages according to the *Separation Of Concerns (SoC)* principle. Later, the Language Renderer only requests the text for an object class and automatically receives the correct translation.

2.4.2. Online Language Resource

If the user requests a language for which no *OLA* exists, the language manager does nothing for the time being. The Google Translate API (Wu et al., 2016; Oster, 2016) was implemented for an online retrieval of translations. An online retrieval takes a lot of resources and costs valuable time. The language manager therefore doesn't translate every prediction (as with offline translations), but only on specific request from the language renderer.

After receiving a request, a work queue job is created. The job consists of the request and a Callback Function. This request is forwarded to the Translation API. Since the translation takes a lot of time compared to offline language resources, the program continues to run in the meantime. Once a translation has been completed, the callback function that was previously packed into the work queue is called. This function passes the determined text to the language renderer. If this translation has taken longer, the Language Renderer has already created the 3D objects with an empty text in the meantime. If it now receives the callback function with the translated text, it changes the text afterwards.

2.5. Language Renderer

The task of the Language Renderer is to create 3D text objects for the two languages and stick them to the given anchor. When the Language Renderer is called, it receives the anchor for the text object (blue sphere in Fig. 7). The provided prediction is an object class identifier. This identifier contains the number of the object class and the corresponding output euron of the object classifier.

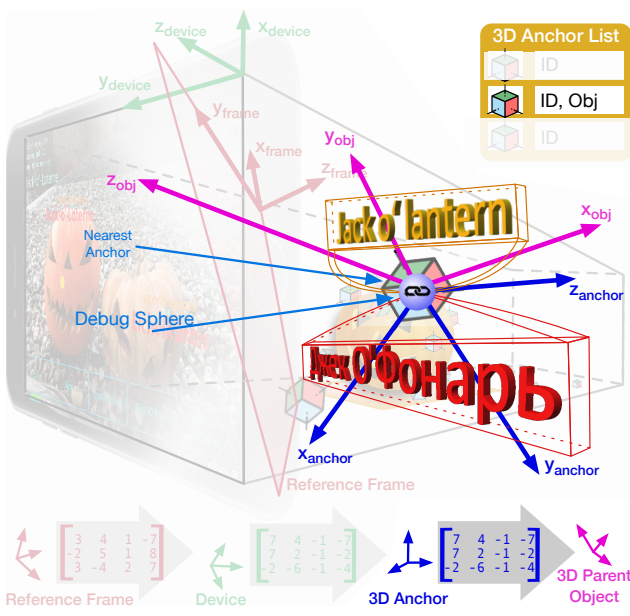


Figure 7: Language Renderer

The Language Renderer now retrieves the texts for the required object class from the Language Manager. According to the *Separation Of Concerns* principle, the renderer does not need to know the language. Four objects will then be created:

- First, there are two text objects that contain the texts of the two languages.

- Furthermore a small *sphereNode* for debugging. It is placed exactly on the anchor to show the user which anchor was selected by the anchor manager.
- Finally a node of dimension 0 to which the 3 objects are added as child objects. This parent node has its own coordinate system and a pivot point in its own coordinate origin. This pivot point is linked to the anchor.

The virtual world knows nothing about the lighting situation in the real world. In order to ensure a consistent appearance of the text objects, the virtual world will be illuminated. This provides plasticity to the objects. The parent node is linked to the anchor. When the world coordinate system is readjusted, all anchors are realigned and updated. The parent node moves along with his updated anchor.

3. Results

With *ARTranslate*, a running software for mobile devices was developed and tested. The source code is publicly available as Open Source (for URL see title page bottom left).

The application was tested with some children, teenagers and adults.

Usability, User AR-Experience After a short explanation, all persons were able to handle it completely. The use, especially by the children, was intuitive.

The global illumination of the virtual world ensures a consistent text look. The texts are easy to read from all positions. Illumination from above gives the text objects plasticity, giving the user the feeling that they fit into the real world. This increases the user acceptance.

The calculation of the 3D point from the 2D screen coordinates using the nearest neighbor method works satisfactorily: the object node is placed on the nearest anchor in the world coordinate system. By chaining the text objects as child nodes of a parent node, which in turn is assigned to an anchor, the texts belonging to an object move synchronously.

After placement, the text objects are located relatively stationary in the room. Movement of the user around the object only induces insignificant position changes.

The frame rate decreases to 13 fps depending on the number of possible objects in the image, see paragraph "Object Recognition".

Session Saving, Object Retrieval The world coordinate system with all its 3D anchor points and object markers can be saved by the user and reloaded later. If the user loads a saved session, the anchor points are still floating and the text objects present in the saved session are not yet present. If the user then walks through the room, the room features are captured. If the room is sufficiently captured, the previously saved text objects appear on the objects. If the user moves on, with every adjustment of the world coordinate system the objects slide closer to the previously stored position.

Language Switching The switching of languages via the chosen rotary control has proven itself and all testers described it as very usable.

During the test phase there were no dropouts or crashes.

Object Recognition The used iPhone 7 achieves a sufficiently high performance with Inception v3 and usually creates 30 - 60 fps in addition to the complete coordinate transformations and the rendering. Many of the everyday objects are recognized and correctly named.

The view usually runs with no noticeable skips. The pipelines for object recognition and the user interface run in parallel. With many objects in the image, the individual predictions of the object recognition algorithm take longer and cannot be compensated on average. The frame rate drops partially to 13 fps.

4. Discussion, Future Work

The users subjectively found it very pleasant that the text objects actually stick to the tapped objects. After a short test period, all users began to move around the objects in order to explore the limits of the system. The teenagers found it funny to walk behind the objects and watch how the texts rotated with them. According to their statements, this significantly increased the quality of the AR Experience, as the objects actually seem to exist in real space.

The evaluation of the survey of the users about the algorithm of the three-dimensional placement of the text objects came to the result that the anchoring of the objects in the users' feeling was at the right location. They stated that they seldom had the feeling that the text objects landed beside their intended destination. In most cases, the dropping point corresponded to the desired or felt target point.

The algorithm which translates the user's typing on the 2D screen into a 3D point with an additional dimension works as expected.

articles before nouns Some language-affine testers reported back to us that the object names of the articles were missing. It has to be said that we used the trained object class names of the Inception v3 network. These are in English and have no articles. The user interface does not display the object classes themselves, but a representation of them from a *Offline Language Array Pointer (OLAP)*, which points to the corresponding *Offline Language Array (OLA)*. At the moment the *OLAs* of e.g. English still contain the same strings as the object classes. The names of the objects in the *OLA* are application-independent interchangeable and we will equip them with articles in the future.

additional language selector The *OLAP* for the language 1 points in the momentary construction stage firmly to the *OLA* for English. This served to check the object classification algorithm in the test phase. The modularization allows the functionality of the *OLA* to be exchanged. We will retrofit a second Language Selector to display any language pairs.

Object Recognition The detection rates are even higher for newer networks. In further enhancements, more powerful hardware will be used to operate detection networks that require more resources.

linguistic scope So far, the app operates on word level. We are planning an enhancement to sentence and text level to give users the possibility "to examine and reflect on language and usage." (Geist and Krafft, 2017). Furthermore, it is

conceivable to implement a speech output. Therewith a written language acquisition is not necessary. The users would receive the speech output acoustically and learn afterwards.

5. Conclusion

The operation area of *ARTranslate* is the interest-driven learning of languages. *ARTranslate* should make it possible to take the first steps towards learning a new language. This is achieved by integrating the discovery of foreign or new language terms into the everyday life of the user. The user is put in a position to discover his environment *immersively* (thaw, 2019). The view through the display offers a world enriched with additional information. By quickly and intuitively switching through the languages, the user discovers representations of his everyday items in many languages. In tests, the Language Selector was often used to find out what for example "Banana" is called in Spanish or French ("Plátano", "Banane"). Or it was astonished that Portuguese and Spanish are not as equal as suspected. A "broom" became Portuguese "vassoura" compared to "cepillo" in Spanish.

Especially children used *ARTranslate* without fear of contact and with a high degree of self-evidence. We recognize a fundamental motivation of users to get knowledge about new languages. The access via Augmented Reality and the integration into their everyday life inspire the users and increase their Motivation.

ARTranslate encourages and supports explorative behaviour and increases interest. For the group of displaced persons or other migrants, it can be a great help to immersively learn names for everyday items in the new environment with a quick glance through the augmented display. The easy access to additional information in everyday activities such as breakfast, cooking, shopping reduces the emotional threshold. This often breaks the first ice and awakens the motivation to continue learning.

We are aware that due to the limited amount of detectable objects the application can be just a first step. We are all convenient. How often do we only lack the first push, the first icebreaker? We want to help make this first step as smooth and simple as possible.

Acronyms

AR	Augmented Reality.
CNN	Convolutional Neural Networks.
OLA	Offline Language Array.
OLAP	Offline Language Array Pointer.
SoC	Separation Of Concerns.

6. Bibliographical References

- Adler, Y. (2011). Kinder lernen Sprache(n): alltagsorientierte Sprachförderung in der Kindertagesstätte. Kohlhammer, Stuttgart. OCLC: 748683717.
- Apple. (2019). Machine Learning.
- ATI. (2019). Mondly. https://www.mondly.com/offer/black-friday-sale?avangate_src=srcpopawards.

- Barreiros, C., Veas, E., and Pammer-Schindler, V. (2016). Pre-attentive Features in Natural Augmented Reality Visualizations. In 2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct), pages 72–73, September.
- Beller, S. and Bender, A. (2010). Allgemeine Psychologie - Denken und Sprache. Bachelorstudium Psychologie. Hogrefe, Göttingen Bern Wien. OCLC: 845680851.
- Bobrovskaya, T. (2019). AR Flashcards - Lernspiele zum Sprachen-Lernen.
- Brandstätter, V., Schüler, J., Puca, R. M., and Lozo, L. (2013). Intrinsische Motivation. In *Motivation Und Emotion*. Berlin, Heidelberg:Springer Berlin Heidelberg, pp. 91–103.
- Dicky, A. and Chuah, K.-M. (2019). Interactive Augmented Reality (AR)-based Learning Cards for Lunbawang Language. In Proceedings of the International University Carnival on E-Learning, October.
- Geist, B. and Krafft, A. (2017). Deutsch als Zweitsprache: Sprachdidaktik für mehrsprachige Klassen. Number 2 in LinguS. Narr Francke Attempto, Tübingen. OCLC: 1001566217.
- Ismail, M. E., Zakaria, A. F., Ismail, I. M., Othman, A., Samsudin, M. A., and Utami, P. (2019). Design and Development of Augmented Reality Teaching Kit: In TVET Learning Context. *International Journal of Engineering & Technology*, 8(1.1):129–134, January.
- Kucuk, S., Yilmaz, R. M., and Goktas, Y. (2014). Effective educational augmented reality applications: Points to consider. In 2014 IEEE 8th International Conference on Application of Information and Communication Technologies (AICT), pages 1–4, October.
- MedAppLab, P. (2019). Marble® - mobile Augmented Reality blended learning environment. <http://marble-app.weebly.com/>.
- Oloruntoba, A. (2019). Consumer Electronics Report 2019 - Telefonie, July.
- Oster, R. (2016). ROGoogleTranslate. <https://github.com/prine/ROGoogleTranslate>.
- Rana, K. and Patel, B. (2019). Augmented Reality Engine Applications: A Survey. In 2019 International Conference on Communication and Signal Processing (ICCSP), pages 0380–0384, April.
- Ruberg, T., Rothweiler, M., and Koch-Jensen, L. (2017). Spracherwerb und sprachliche Bildung: Lern- und Arbeitsbuch für sozialpädagogische Berufe. Ausbildung und Studium. Bildungsverlag EINS, Köln, 2. auflage edition. OCLC: 966421563.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2017). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.
- Schultz, E. (2018). Schweiz - Smartphonebesitz 2018. <https://de.statista.com/statistik/daten/studie/977767>.
- Sirakaya, M. and Alsancak Sirakaya, D. (2018). Trends in Educational Augmented Reality Studies: A Systematic Review. *Malaysian Online Journal of Educational Technology*, 6:60–74, April.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2818–2826, Las Vegas, NV, USA, June. IEEE.
- Tenzer, F. (2019). Smartphone-Besitz - Kinder und Jugendliche in Deutschland 2019. <https://de.statista.com/statistik/daten/studie/1106/umfrage/handybesitz-bei-jugendlichen-nach-altersgruppen/>.
- thaw. (2019). Wie geht das? Immersives Englisch-Lernen – ELBINSELSCHULE.
- Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv:1609.08144 [cs]*, October.

7. Language Resource References

- Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv:1609.08144 [cs]*, October.