



ugr

Universidad
de Granada

COMPUTACIÓN UBICUA E INTELIGENCIA AMBIENTAL
GRADO EN INGENIERÍA INFORMÁTICA

ARWrite

Autor

Carlos Sánchez Páez



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

CURSO 2019-2020

Índice

1. Descripción de la app	2
2. Motivación	3
2.1. Problema a resolver	3
2.2. Competencia	3
2.3. Opiniones de expertos	4
3. Recursos a utilizar	5
4. Descripción detallada de la aplicación	5
5. Aprendizaje del entorno	8
6. Informe económico	9
7. Entrevistas a expertos	9
Referencias	10

1. Descripción de la app

El objetivo de este proyecto es desarrollar una aplicación basada en realidad aumentada que se utilizará para reforzar la enseñanza del trazo y la caligrafía a niños/as de infantil. Ésto se conseguirá haciendo que el usuario (niño/a) intente dibujar una letra (elegida por el profesor/a) en el aire con un puntero. El sistema identificará mediante técnicas de OCR la entrada e informará si coincide con lo esperado.

La aplicación constará de dos ventanas, una en la que se verá la letra objetivo y otra que mostrará la captura de video y el trazo dibujado hasta el momento.

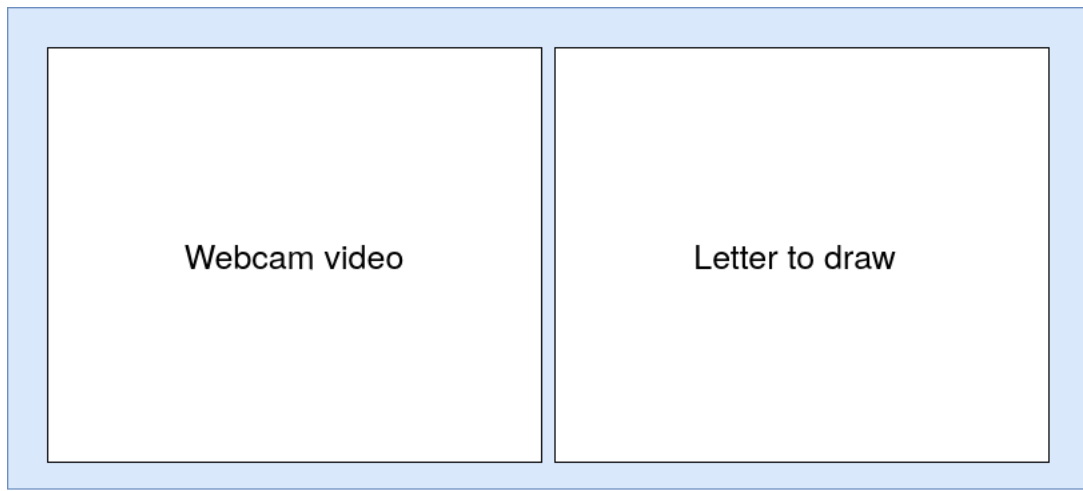


Figura 1: Diseño de la aplicación

2. Motivación

2.1. Problema a resolver

La tecnología tiene un rol clave hoy día en prácticamente todos los ámbitos de nuestra vida. En el educativo, las clases cada vez se enriquecen más gracias a recursos digitales. Esto ocurre en prácticamente todos los sectores, desde el infantil hasta las altas titulaciones.

En el caso de las primeras etapas escolares de los alumnos (infantil), la enseñanza apoyada por medios digitales se hace cada vez más necesaria. En (Robles-Meléndez, 2012) se justifica su uso en el aula ya que potencian la motivación y comprensión de la materia para el alumno. Podemos contrastar la efectividad de las mismas en publicaciones como (Bonneton-Botté y cols., 2020), que propone una aplicación para aprender a escribir en una tablet.

En este proyecto se pretende aprovechar la serie de ventajas que trae la tecnología al aula para conseguir que el alumno/a desarrolle el aprendizaje del trazo de una forma motivadora y amena, cambiando la tradicional escritura en pizarra por escritura en realidad aumentada.

2.2. Competencia

En la actualidad ya existen aplicaciones de realidad aumentada orientada a la educación de preescolar. Por ejemplo, QuiverVision (Apple, 2016) ofrece una aplicación en la que el usuario colorea unas plantillas con dibujos y al enfocarlas con una tablet o smartpho-
ne, el dibujo se vuelve interactivo, ofreciendo al alumno/a jugar con el personaje que ha dibujado. Math Alive (AliveStudios, 2013) propone juegos que consisten en colocar cartas frente a una cámara para aprender a contar y realizar operaciones básicas.

Existen más aplicaciones de realidad aumentada enfocadas a la educación en preescolar, pero se centran en que el alumno/a rellene fichas y las enfoque con la cámara para hacerlas interactivas o bien. Aquí se encuentra el elemento diferenciador del proyecto: el usuario realizará el trazo frente a la pantalla y la aplicación le proporcionará un *feedback*.



Figura 2: *Quiver* en funcionamiento

2.3. Opiniones de expertos

En la sección 2.1 podemos consultar varias referencias bibliográficas que avalan la efectividad de las aplicaciones de realidad aumentada en el ámbito escolar. Además, contamos con la opinión de Isabel María Páez Alba, diplomada en Educación General Básica en la especialidad de filología francesa por la Escuela Universitaria María Inmaculada de Antequera (Málaga) en el año 1988 y que lleva ejerciendo como profesora de educación infantil durante 31 años:

“

La Ley Orgánica 2/2006, de 3 de mayo, de Educación y el Decreto 428/2008, de 29 de julio, establecen la ordenación y las enseñanzas correspondientes a la Educación Infantil en Andalucía. La Educación Infantil es concebida como una etapa única en la que se inicia a los niños y niñas de 3, 4 y 5 años en determinados aprendizajes. Por ello, en esta etapa, la ley propone un acercamiento a la lectoescritura. La escritura sigue siendo un aprendizaje básico para toda persona. Es la forma en la que podemos comunicarnos, expresar nuestros sentimientos, intercambiar opiniones; en definitiva, es la forma en la que podemos transmitir. Es en esta etapa donde van a tener su primer contacto con el lenguaje escrito; es un reto arduo al que los niños y niñas se enfrentan. Antes de escribir es necesario que el niño domine y controle sus movimientos, que sea capaz de desplazar la mano en el sentido deseado; debe adquirir habilidad grafomotriz. Todo esto se adquiere de forma lúdica y activa, principios metodológicos básicos y fundamentales en la Educación Infantil.

El proyecto presentado por D. Carlos Sánchez Páez aúna cuatro aspectos relevantes en el aprendizaje lectoescritor. Por un lado, va a iniciar al niño/a en el proceso de la escritura. Por otro, va a dotar al alumnado de un soporte motivador, innovador, atractivo para los niños y niñas. Las nuevas tecnologías son tremendamente atractivas para esta generación de alumnos que han nacido en esta era. En tercer lugar, va a favorecer la autocorrección y autoevaluación. El niño/a va a ser artífice de su propio aprendizaje. Y por último y no por eso menos importante, ofrece a los niños y niñas los principios metodológicos fundamentales para el aprendizaje en su corta edad: actividad y juego. En definitiva, se aglutina en este proyecto toda la base de una enseñanza completa, atractiva, divertida y personalizada.

”

3. Recursos a utilizar

Para el desarrollo del proyecto se pretenden utilizar las siguientes tecnologías:

- **Plataforma destino:** PC
- **Lenguaje de programación:** Python (Van Rossum y Drake, 2009)
- **Frameworks**
 - OpenCV (Bradski, 2000). Librería orientada a la visión por computador en tiempo real.
 - Keras (Chollet y cols., 2015). Librería orientada a redes neuronales. Se utilizará junto a MNIST (LeCun y Cortes, 2010), una base de datos de dígitos escritos a mano para construir una red neuronal que clasificará la entrada del usuario.

4. Descripción detallada de la aplicación

La aplicación seguirá el siguiente esquema:

1. La aplicación muestra un cuadro de dialogo solicitando una letra objetivo.
2. El usuario elige la letra objetivo.
3. La ventana principal de la aplicación se abre. En el lado izquierdo se ve la captura de la cámara y en el derecho la letra elegida por el usuario en fuente escolar.
4. El usuario toma su marcador y dibuja la letra en el aire. Cuando termine, deberá ocultar la parte azul del marcador a la cámara.
5. La aplicación procesa el trazo y lo envía a la red neuronal previamente entrenada para obtener una predicción.
6. Si la predicción es correcta, se le preguntará al usuario si desea jugar otra vez. En caso contrario, se informará de la letra que ha dibujado y se le pedirá que lo intente de nuevo.

En esta aplicación se ha implementado el uso de un marcador de color azul, como por ejemplo un rotulador de pizarra con el tapón puesto. El resto del cuerpo del marcador deberá ser de otro color para que al girarlo se oculte.



Figura 3: Ejemplo de marcador válido

Junto a la aplicación en sí, se adjunta un ejecutable para distribuciones Linux que instala la aplicación y sus dependencias además de crear las correspondientes entradas en el menú de aplicaciones (aplicación y desinstalador).

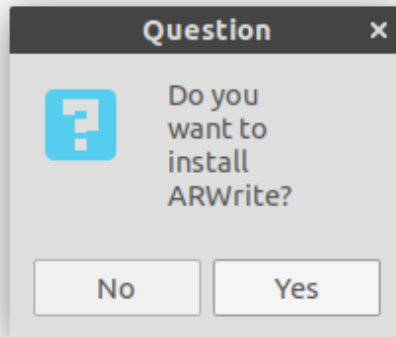


Figura 4: Instalador de ARWrite

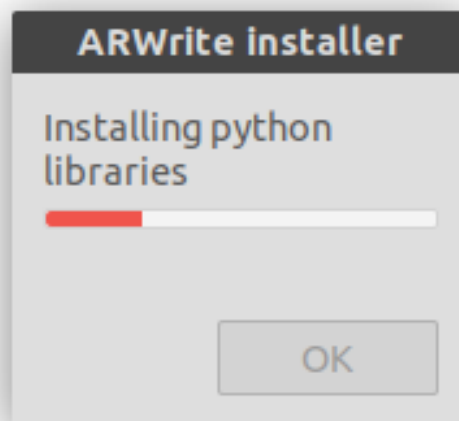


Figura 5: Instalador de ARWrite (2)

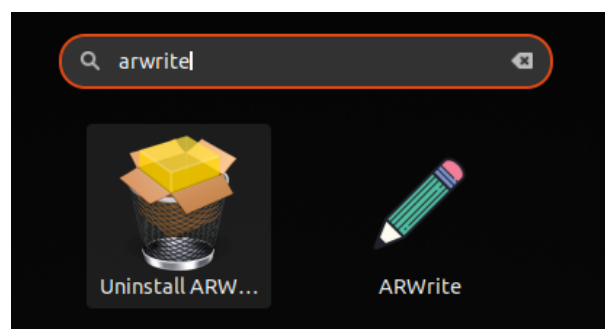


Figura 6: Entradas en el menú de aplicaciones

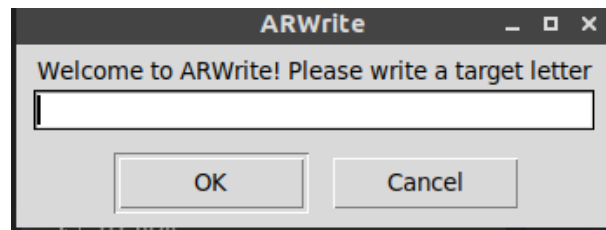


Figura 7: Solicitud de letra objetivo

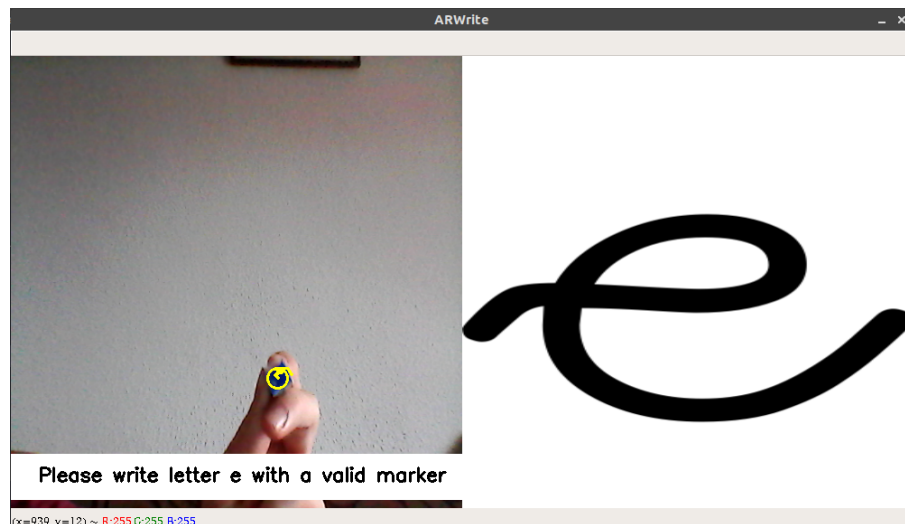


Figura 8: Ventana principal de la app

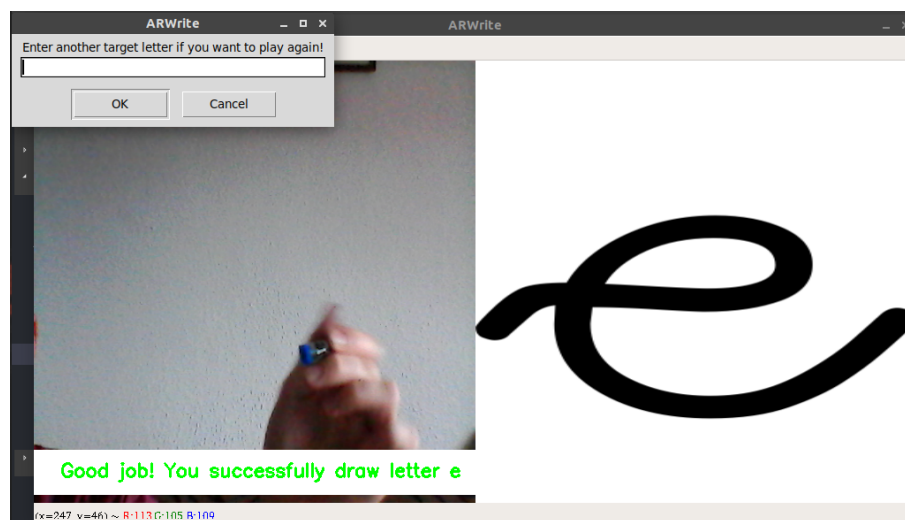


Figura 9: Letra dibujada correctamente

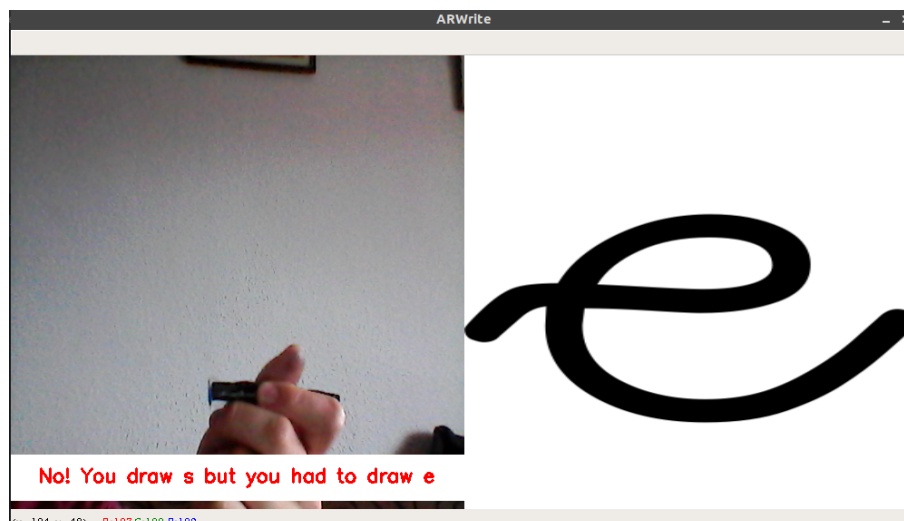


Figura 10: Letra incorrecta dibujada

5. Aprendizaje del entorno

Durante el diseño y la implementación de la aplicación los entornos utilizados fueron los siguientes:

1. Frameworks para el desarrollo y entrenamiento de la red neuronal

- a) Keras (Chollet y cols., 2015). Es una librería escrita sobre TensorFlow (Abadi y cols., 2015) que permite la implementación a alto nivel de redes neuronales. En la aplicación se utilizarán redes neuronales de tipo convolucional, con las que ya he trabajado antes (<https://github.com/csp98/image-recognition>)
- b) Numpy (Oliphant, 2006). Biblioteca para procesamiento científico de datos. Es necesaria para trabajar con varias dimensiones en el desarrollo de la red neuronal.

2. Framework para realidad aumentada: OpenCV (Bradski, 2000). Usada para capturar la imagen desde la cámara, identificar el marcador, dibujar los trazos en la pantalla, etc. Es la piedra angular de la aplicación.

3. Frameworks para manejar interfaces gráficas

- a) Tkinter (Hughes, 2000). Biblioteca de Python (Van Rossum y Drake, 2009) que utiliza para mostrar la ventana en la que se solicita la letra objetivo al usuario.
- b) Zenity (Foster, s.f.). Aplicación de Linux que se usa para crear la interfaz gráfica del instalador y desinstalador.
- c) Pkexec (Zeuthen, s.f.). Utilidad gráfica de Linux para solicitar permisos de superusuario. Se usa en el instalador.

6. Informe económico

ARWrite se distribuye bajo una licencia *GNU General Public License (GNU General Public License, version 3, 2007)*, lo que implica que el usuario podrá utilizar este software siempre y cuando **no sea con fines lucrativos**. La decisión de que este software sea gratuito viene propiciada por el estado de desgaste en el que se encuentra el sector educativo, siendo cada vez más los recortes que se le aplican, el ratio de alumnos por aula, etc., lo que hace que la enseñanza sea cada vez más complicada.

Como los componentes de *ARWrite* son gratuitos, el coste para el desarrollo de la aplicación ha sido nulo. Esto hace que el período de amortización sea inexistente, ya que no se ha invertido dinero en desarrollo. El único aspecto en el que se ha necesitado invertir es en la construcción de un marcador válido (por ejemplo, un rotulador con capuchón azul) (en torno a unos 2-3€). En mi caso, construí este marcador mediante un destornillador y un trozo de plástico azul pegado a la punta, ya que el estado de alarma hizo imposible la adquisición del rotulador.

Un posible modelo de negocio a aplicar sería una versión *Premium* de *ARWrite*. Por ejemplo, se podría añadir un panel de configuración para que el usuario (profesor/a en este caso) configurase el color del marcador a su gusto, añadir la funcionalidad de aprender a dibujar los números, símbolos, frases, etc. Esta versión *Premium* se podría distribuir de dos formas diferentes:

- Pago por versión. Cada nueva versión tendrá un coste fijo (en torno a 10€).
- Suscripción. Los usuarios se suscribirían a *ARWrite* por 5 al mes y se les entregaría una nueva versión del software cada mes.

7. Entrevistas a expertos

Debido a la situación actual causada por el COVID-19, no ha sido posible poner a prueba *ARWrite* en un escenario real, ya que a partir de marzo se suspendieron las clases presenciales en todos los centros educativos.

Enlace al código:

<https://drive.google.com/drive/folders/14e1B9ZBljGMUnRDmYnQP7xZVFZ0HRc0r?usp=sharing>

Referencias

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2015). *Tensorflow: Large-scale machine learning on heterogeneous systems*. <https://www.tensorflow.org/>.
- AliveStudios. (2013). *Math alive*. <http://www.quivervision.com>.
- Apple. (2016). *Quivervision 3d*. <http://www.quivervision.com>.
- Bonneton-Botté, N., Fleury, S., Girard, N., Le Magadou, M., Cherbonnier, A., Renault, M., ... Jamet, E. (2020). Can tablet apps support the learning of handwriting? an investigation of learning outcomes in kindergarten classroom. *Computers and Education*.
- Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- Chandra, A. L. (2018). *Alphabet recognition through gestures*. <https://towardsdatascience.com/tutorial-alphabet-recognition-deeplearning-opencv-97e697b8fb86>.
- Chollet, F., y cols. (2015). *Keras*. <https://keras.io>.
- Foster, G. (s.f.). *Zenity*. <https://help.gnome.org/users/zenity/>.
- Gnu general public license, version 3*. (2007). <http://www.gnu.org/licenses/gpl.html>.
- Hughes, P. (2000). *Python and tkinter programming*. <https://pypi.org/project/tk-tools/>.
- LeCun, Y., y Cortes, C. (2010). *Mnist handwritten digit database*. <http://yann.lecun.com/exdb/mnist/>. Descargado de <http://yann.lecun.com/exdb/mnist/>.
- Oliphant, T. E. (2006). *A guide to numpy*. <https://numpy.org/>.
- Robles-Meléndez, W. (2012). Tecnología en el aula infantil. apuntes y comentarios. *Revista Complutense de Educación*, 23(1), 149-160. Descargado de <https://revistas.ucm.es/index.php/RCED/article/view/39107>.
- Van Rossum, G., y Drake, F. L. (2009). *Python 3 reference manual*. Scotts Valley, CA: CreateSpace.
- Zeuthen, D. (s.f.). *pkexec*. <http://manpages.ubuntu.com/manpages/trusty/man1/pkexec.1.html>.